

Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965
PL ISSN 0860 – 1674

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH NR 2(26) LUTY 1988 CENA 100ZŁ

AMSTRAD INFO

MUZYKA WŁASNA, NUTY KOMPUTERA



DMP3160

MIGAJĄCY KURSOR

BUFORY

INWAZJA WIDEFONÓW

SABOTEUR II

NA GORĄCO

Prawie równocześnie z ukazaniem się tego, lutowego numeru naszego pisma, znajdzie się też na rynku numer specjalny "Bajtka" poświęcony wyłącznie komputerom Atari. "BAJTEK-ATARI" jest odpowiedzią na prośby naszych Czytelników, a jednocześnie zapoczątkowuje całą serię dodatków specjalnych przeznaczonych dla użytkowników poszczególnych typów komputerów osobistych dostępnych na naszym rynku.

Od dawna otrzymywaliśmy listy, których nadawcy mieli do nas pretensje o to, że kolumn klanowych, poświęconych temu właśnie komputerowi, który oni mają, jest za mało. Świadczyło to częściowo o nieuważnym czytaniu innych rubryk, dotyczących spraw wspólnych dla wszystkich typów komputerów, ale z drugiej strony stanowiło też dla nas wyraźny sygnał, że potrzebne jest na rynku większe zróżnicowanie periodycznych wydawnictw komputerowych. W swoim czasie zjawisko takie miało już miejsce na zachodnich rynkach pism komputerowych, nie ma więc nic dziwnego w tym, że powtórzyło się również u nas. Tam pojawiły się w odpowiedzi na to zapotrzebowanie pisma typu "Amstrad-User" czy "Spectrum-User" — u nas, mamy nadzieję, takim odpowiednikiem będą monotematyczne wydania dodatkowe "Bajtka" adresowane do użytkowników komputerów Atari, Commodore, Amstrada i Spectrum.

Wkrótce po "ATARI-BAJTKU" powinien pojawić się w kioskach Ruchu "BAJTEK-COMMODORE", następnie kolejny "BAJTEK-ATARI" (jako że ten typ komputera zdominował nasz rynek), a potem przyjdzie kolej na inne komputery. Ponieważ są to wydania dodatkowe, więc nie obejmuje ich prenumerata — przypominam o tym, aby nie było nieporozumień.

Oczywiście, to swoiste rozrośnięcie się i usamodzielnienie kolumn klanowych nie oznacza wcale, że zrezygnujemy z nich na łamach "Bajtka". Mają tu one swoje stałe miejsce — z tym że będziemy starali się prezentować w nich materiały najatrakcyjniejsze, mogące zainteresować również osoby dysponujące innymi komputerami. Będziemy zresztą starali się specjalnie tę uniwersalność materiałów zamieszczanych na kolumnach klanowych podkreślać i wskazywać.

A teraz kilka uwag na gorąco z trwających właśnie targów — wystawy "Komputer 88". To co się dzieje w momencie pisania tych słów na pierwszych czterech piętrach warszawskiego Pałacu Kultury i Nauki zaskoczyło wszystkich. Impreza, która odbywa się w Polsce zaledwie po raz trzeci już stała się instytucją tak znaczącą, że trudno sobie nawet wyobrazić bez niej lutowy pejzaż stolicy. Wykorzystana do ostatniego metra kwadratowego powierzchnia targowa, zaaferowane twarze oferentów i potencjalnych nabywców, tłumy "gości zaproszonych" od chwili otwarcia do zamknięcia wystawy — to najlepiej świadczy o sukcesie.

Dzięki strategicznym decyzjom podjętym w sztabie ludzi odpowiedzialnych za reformę gospodarczą powstał błyskawicznie w Polsce rynek komputerowy z prawdziwego zdarzenia. Powtarzam — nie wbrew reformie, jak niektórzy lubią narzekać, tylko właśnie w wyniku i w duchu reformy! Dzięki temu dziesiątki, a w tej chwili już nawet setki małych i średnich firm komputerowych mogło na serio, z nadzieją na niezłe zyski, zająć się informatyzacją poszczególnych dziedzin naszej gospodarki. Warszawska wystawa była właśnie przeglądem najprężniejszych w tej dziedzinie. Trzeba przyznać, że był to przegląd imponujący.

Okazało się przy okazji, że wykształcił się na styku sektora zagraniczno-prywatnego z państwowym i spółdzielczym specyficzny typ polskiego menadżera branży komputerowej. Zdominowali warszawską imprezę młodzi ludzie między 20 a 30 rokiem życia. Dobrze władający w większości językiem angielskim, otwarci w kontaktach, precyzyjnie w kalkulacjach a jednocześnie umiejący szy bko podejmować decyzje. W nich nasza nadzieja!

Szkoda tylko, że tak trudno było się dostać na "Komputer 88" ich młodszym kolegom — uczniom szkół podstawowych i średnich. Do godziny 16.00 impreza była przeznaczona tylko dla "gości zaproszonych", dopiero między 16.00 a 18.00 mogli przez trzy dni obejrzeć ją "normalni" ludzie.

Te sześć godzin — i to w nienajlepszej porze, bo dawno po zajęciach szkolnych — to było wszystko, na co mogli liczyć warszawscy uczniowie. Wycieczki z innych miast były zupełnie bez szans!

Można zrozumieć motywacje organizatorów i wystawców, chcących przede wszystkim "robić biznes", ale trudno zgodzić się na traktowanie po macoszemu potrzeb tych, którzy do "biznesu" będą jeszcze przez kilka lat dorastać. Największa wystawa komputerowa w Polsce powinna również w szerszym stopniu spełniać funkcje edukacyjne, powinna stwarzać młodzieży możliwość choćby "napatrzenia się" na najnowszą technikę komputerową. Takie są potrzeby społeczne i organizatorzy nie powinni ich w żadnym razie lekceważyć! Sześć godzin dla szerokiej publiczności to stanowczo za mało. Redakcja "Bajtka" wyraża nadzieję, że w programie przyszłorocznej imprezy w szerszym stopniu uwzględnione zostaną interesy szerokich rzesz młodzieży!

Tyle impresji na gorąco. Pełna relacja z "Komputera 88" — już w następnym numerze "Bajtka"..

Waldemar Siwiński



SZANOWNY PANIE REDAKTORZE!

Po publikacji na temat Warsaw Basic w Bajtku nr 7 z tego roku utwierdziło się moje przekonanie, że sprawa edukacji naszej młodzieży w zakresie informatyki innych przedmiotów w oparciu o mikrokomputery nie może być dłużej zaniedbywana, leżąc ciągle w sferze planów i badań bez realizacji, tym bardziej, że obecnie już posiadamy wszystko, co jest potrzebne do takiego przedsięwzięcia, a więc polski interpreter - program Warsaw Basic, czekające moce produkcyjne do jego produkcji i pewną ilość komputerów w szkołach i na wyższych uczelniach.

Szkoły nasze obecnie są w nikłym procencie wyposażone w jakiekolwiek komputery. W tej nikłej ilości występują takie typy jak Commodore, Atari, Spectrum Amstrad i polski Mentum. Polski Junior 800 dopiero będzie wprowadzony.

Taka różnorodność i niedostatek uwarunkowane są sytuacją ekonomiczną i rynkową naszego kraju jest jak mi się wydaje w pewnym sensie powodem opóźnienia w zasadniczych decyzjach i realizacjach w zakresie edukacji informatycznej i innej w oparciu o komputery naszej młodzieży.

Ta różnorodność i wypływające z tego niezdecydowanie oraz żywiołowe edukowanie się naszej młodzieży nie jest najlepszym rozwiązaniem. Można przecież w tej sytuacji zaproponować strefowe oparte na systemie wojewódzkim wyposażanie szkolnictwa w dostępne różne komputery.

Natomiast w drugim etapie po okrzepnięciu naszego przemysłu mikrokomputerowego, zdobyciu doświadczenia w szkołach i zużyciu się obcych maszyn cyfrowych wprowadzić generalnie nasz np. Junior 800 standardowo wyposażony w Warsaw Basic. Zresztą ten system wyposażania komputerów jednocześnie przy sprzedaży w programy jest powszechnie stosowany.

W historii innych dziedzin naszego żyda ekonomicznego wiele mieliśmy przykładów podobnych rozwiązań, jak zaproponowane powyżej. W obecnej naszej sytuacji pozwoli to na szybkie skasowanie opóźnienia edukacyjnego naszej młodzieży w tym zakresie.

W oparciu o powyższe kompletnym nieporozumieniem byłoby gdybyśmy tylko w oparciu o obce i obcojęzyczne programy wychowali naszą młodzież tym bardziej, że swój i to jeden z najlepszych interpretatorów-programów WSO Warsaw Basic piszący po polsku i rosyjsku mamy. Został on opracowany przez dr Krzysztofa Gajewskiego i doc. dr hab. Bogusława Radziszewskiego z IPPT PAN w Warszawie, w ramach pracy nad Warszawskim Systemem Operacyjnym. Ten interpreter-program został zbudowany na najużyteczniejszej i najpopularniejszej na świecie strukturze Basic tak jak kiedyś nasz język na strukturze gramatyki łacińskiej.

Elementy edukacyjne W.B. w zakresie szkolnej informatyki są następujące:

1. Nauka programowa i jego zaawansowanych elementów na bazie Basic (p. Bajtek) biblioteka podprogramów, programowanie proceduralne i strukturalne.
2. Geometria, obrazy i figury, płaskie przestrzenne i ich obroty.
3. Język polski na ekranie i na papierze.
4. Praca na komputer w dowolnym języku (ekran papier) w tym wydaje się być bardzo interesującą propozycja pracy w języku rosyjskim, szczególnie w początkowych klasach nauczania.

Włączenie mikrokomputera w ten proces może znacznie przyspieszyć i podwyższyć efektywność nauczania języków obcych.

5. W.B. daje możność efektywnej nauki fizyki, chemii, matematyki, historii i innych przedmiotów w języku polskim.

Ten bezsprzecznie jeden z najlepszych na świecie interpretatorów jak uznaje to opinia środowiskowa informatyków powinien być jak najszybciej wprowadzony na wyposażenie szkolnictwa. Na razie naturalnie do szkół posiadających mikrokomputery typu Commodore, gdyż na ten typ komputera został on w pierwszym rzędzie opracowany. Po zapadnięciu długo oczekiwanych, co do wyposażenia szkolnictwa w ten język—decyzji może on być szybko dostosowany do każdego innego komputera w tym także istniejących w Wyższym Szkolnictwie IBM-ów.

W sumie wprowadzenie WSO i WB do naszego systemu edukacyjnego niższego i średniego szczebla a po przystosowaniu WB do IBM-ów także do Szkolnictwa Wyższego wydaje się być już palącą koniecznością a nie jedną z alternatyw.

Dr inż. Krzysztof August Kunert
Warszawa

"BAJTEK" — MIESIĘCZNY DODATEK DO "SZTANDARU MŁODYCH" ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21 -12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański- -redaktor naczelny "Sztandaru Młodych".

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego "SM" — kierownik zespołu "Bajtka"), Roman Poznański (z-ca sekretarza redakcji "SM" — sekretarz zespołu "Bajtka"), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Gogolewski, Andrzej Kowalewski, Andrzej Podulka, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski.

Klany redagują: Commodore — Klaudiusz Dybowski, Amstrad-Schneider — Tomasz Pyć, Sergiusz Wolicki, Spectrum — Marcin Przasnyski, Michał Szuniewicz, Atari — Wiesław Migut, Wojciech Zientara.

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wóltańska

WYDAWCA: RSW "Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414. Cena 100 zł.

Skład techniką CRT-200, przygotowalnią offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW "PRASA- -KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51. Nr zlecenia 010188 n. 150.000 egz. U-113



Bajtek

MUZYKA WŁASNA NUTY KOMPUTERA

Rozmowa z kompozytorem i muzykiem — **Krzysztofem Sadowskim**



Często kiedy patrzę na zapis nutowy swojej kompozycji, jestem zdumiony...

— Grasz na komputerze?

— Tak

— Od kiedy?

— Od paru miesięcy. Instrumenty elektroniczne pokazały się też u nas. Pod tym względem nie odstajemy od reszty świata.

— Jak się gra?

— Dobrze. Ale ty zadajesz mi pytania, na które mogę odpowiadać stereotypami. Dlatego pozwól, że przedstawię trochę teorii. Obecnie muzyka jazzowa nie cieszy się tak wielką popularnością jak kiedyś. Dlatego utrzymywanie dużego zespołu staje się trudne.

— Komputer jest dobry na kryzys?

— Tak. Mogłem nabyć kilka instrumentów, które w zasadzie zastępują orkiestrę. Współczesna elektronika doprowadziła do tego, że można imitować głos każdego instrumentu. Ba, tworzyć nawet nowe barwy, które do tej pory nie były znane.

— Naprawdę?

— Dzięki komputerowi można nawet rozkładać poszczególne dźwięki na różne wysokości — jest to szczególnie pomocne w czasie komponowania nowych utworów. Zapamiętuje wszystko to co ja zagram. Działa równocześnie tak jak magnetofon. Na jednym śladzie nagrywam perkusję, na drugim kontrabas, na trzecim smyczki, na czwartym gitarę, na piątym fortepian, na szóstym trąbki, na siódmym jakieś instrumenty syntezatorowe. Komputer to wszystko razem miesza i po chwili mogą odtworzyć i przesłuchać nowy skomponowany przeze mnie utwór.

— Z komputerem pod pachą biegasz więc na koncerty?

— W zasadzie mogę. Ale normalnie przegrywam skomponowany utwór na taśmę magnetofonową, zanoszę do radia i wielu słuchaczy może w ten sposób nagrane moje kompozycje usłyszeć w programie I i III. W podobny sposób komponuję podkład muzyczny do filmów.

— Czy w czołówce filmowej podajesz również jako współautora nazwę komputera?

— Nie. Kompozycja jest przecież moja. Została przez komputer tylko wzbogacona poprzez rozłożenie dźwięków takie jakie ja wymyśliłem.

— Jak więc pracujesz?

— Całą aparaturę mam w domu. Miałem sporo różnego rodzaju pomysłów muzycznych, których z braku czasu nie mogłem zrealizować. Teraz mam czas, na przykład późnym wieczorem. Siadam, komponuję. Słucham. Jeśli coś mi się nie podoba to mogę poprawić, cofnąć, zmienić. Pozamieniać głosy, pozamieniać takty, powtórzyć, a kiedy jest to zupełnie niedobre kasuję. Dzieje się to tak niby jak na magnetofonie, ale na magnetofon nagrywałem cały utwór i mogłem go najwyżej skasować. Tu jest inaczej. I to jest sukces nowoczesności.

— Twoją pracę można było oglądać na wystawie, którą zorganizowała firma Atari w czasie festiwalu Jazz Jamboree.

— Tak. Na komputerze Atari pokazywałem swoją pracę. Wielu muzyków, którzy oglądali pokaz było zachwyconych możliwościami komponowania przy pomocy komputera. Fascynowało ich to, że nuty, które widoczne były na ekranie można było przy pomocy myszki poprawiać i zmieniać, zwiększać lub zmniejszać ich wartości. Jest to naprawdę prawdziwa rewolucja muzyczna i niezwykła pomoc dla kompozytorów.

— Z pewnością niedługo sam komputer będzie komponował nowe utwory!

— Nie. Często spotykam się z zarzutem, że komputer tworzy muzykę nieludzką, niehumanistyczną, muzykę dla robotów. Nieprawda. Komputer odtwarza tylko to co ja zagrałem. Odtwarza to w sposób idealny.

— Tylko?

— Kiedyś, kiedy nie pracowałem z komputerem i komponowałem muzykę dla swojego zespołu to grałem ją na fortepianie, zapamiętywałem i potem dopiero zapisywałem prostymi nutami na papierze. Natomiast obecnie pracując z komputerem mam, jak już mówiłem, ułatwione zadanie. Komputer jednocześnie zanotuje to wszystko na papierze, to co zagrałem. Często, kiedy patrzę na zapis nutowy swojej kompozycji, jestem zdumiony, że napisałem taki skomplikowany utwór. Nieraz wydaję komputerowi rozkaz by uprościł zapis, żeby nie było tyle łuków, tyle szesnastek czy trzydziestodwójek tylko same ósemki lub ćwiartki. I komputer robi to.

— Jak widzę zaprzyjaźniłeś się z komputerem.

— Jest to istotnie rzecz fascynująca. Jeszcze raz powtarzam, że nigdy bez ułatwień jakie daje komputer tak wiele nie skomponowałbym w ostatnim czasie. Program opracowany na Atari pozwala na wiele.

— Z tego wynika, że ja nie zostanę kompozytorem. A myślałem, że kupię sobie komputer oraz programy, usiądę i zacznę grać.

— Prymitywnym kompozytorem możesz zostać. Jeśli pomylisz prostą frazę ze trzy razy i każesz komputerowi wyrównać i poprawić błędy i potem każesz powtórzyć to dziesięć razy z różnego rodzaju instrumentami perkusyjnymi to być może że uda ci się skomponować jakiś utwór, który można zaprezentować w dyskotece. Tylko, że z mojego punktu widzenia będzie to muzyka nie ludzka i bezwartościowa. Ale idąc za tokiem twego rozumowania na dobrą sprawę dzięki komputerom każdy może zostać kompozytorem.

— No właśnie...

— Tak samo każdy może zostać wirtuozem. No, może nie każdy, bo jeśli w ogóle nie gra na żadnym instrumencie, to nie mógłby. Ale jeśli już gra, nawet bardzo powoli jakąś melodię, lub jakąś frazę a potem puścić to szybciej, i naprawdę będzie to wykonanie, którego nie powstydzi się zawodowy muzyk.

— I nie widzisz w tym żadnego niebezpieczeństwa?

— Tak. W tym tkwi pewne niebezpieczeństwo. I dużo właśnie takiej tandety zaczyna się szerzyć w dyskotekach na Zachodzie. Ja liczę na to, że nawet człowiek, który na co dzień posługuje się komputerem, zachowuje w swojej pamięci prawdziwą muzykę — Chopina, Pendereckiego...

Myślę, też, że najbardziej interesującym zadaniem tych, którzy programują jest tworzenie oryginalnego brzmienia instrumentów, które znamy od setek lat, skrzypiec czy wiolonczeli. I z tego najlepiej wynika, że człowiek nie zatracił jeszcze poczucia rzeczywistych dźwięków, które płyną z natury. Naturalny dźwięk naturalnego instrumentu z komputera. To jest to o co biją się najlepsi programiści.

Rozmawiał: Kazimierz Treger

GRA O JUTRO

KOMPUTERZE – WSZYSTKIEGO NAJLEPSZEGO!

Nie będziemy drugą Japonia. Nie zapowiada sie nawet byśmy mieli ambicje i możliwości stać sie drugim Tajwanem — przynajmniej, jeśli wziąć pod uwagę rolę, jak w gospodarce obu krajów zamierza powierzyć sie informatyce i przemysłowi produkującemu na jej potrzeby. Dla nas oznacza to. że długo jeszcze w "Bajtku" nie pojawia sie klany komputerów o nazwach bardziej swojskich niż "Atari" czy "Commodore".

Do takich, niezbyt dla fanatyka komputeryzacji optymistycznych, wniosków skłania lektura przyjętego w końcu 1987 roku przez Komitet d/s Nauki i Postępu Technicznego "Programu rozwoju informatyki do roku 1995". Warto jednak poznać jego założenia. Choćby dlatego, że, jak zapewniają autorzy, jego założenia stanowią maksimum tego, co można osiągnąć w roku 1995.

Co konkretnie przewiduje "Program..."? Czy ci, którzy m.in. dzięki "Bajtkowi" pokonują pierwsze szczeble informatycznego wtajemniczenia będą mogli w przyszłej pracy zawodowej wykorzystać nabyte wiadomości?

Program, nieprzypadkowo chyba wybija na plan pierwszy bardzo konkretne zastosowania informatyki jako narzędzia dla zwiększenia efektywności całej gospodarki. Dlatego też przewiduje się przede wszystkim zastosowanie komputerów w systemach wspomagania prac inżynierskich i projektowych, badań naukowych a także procesów produkcyjnych i pomiarowo-kontrolnych. Każde biuro projektowe połowy lat dziewięćdziesiątych dysponować ma systemem komputerowym. Również do wszystkich nowocześniejszych zakładów wprowadzone być mają systemy CAM, zaś licencje produkcyjne nabywane będą także łącznie z odpowiednim hardwarem i softwarem.

Dla realizacji tylko tych trzech zadań potrzeba będzie ok. 250 tys. mikrokomputerów. 550 tys. dalszych otrzymać powinna oświata i szkolnictwo wyższe, a 40 tys. służba zdrowia. W sumie w Polsce połowy lat dziewięćdziesiątych pracować ma prawie pół miliona profesjonalnych mikrokomputerów, drugie tyle edukacyjnych i 400 tysięcy domowych. Będzie zatem z czego korzystać i w szkole, i w pracy. Tyle, że taki poziom rozwoju informatyki oznacza postawienie się na równi z krajami kapitalistycznymi w przypadku... zatrzymania na 8 lat wskazówek zegara.

Najwięcej kłopotów sprawia i sprawiać będzie jeszcze długo technika i technologia. Program zakłada wprawdzie wprowadzanie do produkcji coraz to nowszych typów mikrokomputerów, ale "pewniaków" jest w nim tylko dwóch: "Mazovia-1016" oraz "Elvro-800". Zważywszy na poziom nowoczesności obu tych konstrukcji wydaje się nieporozumieniem zakładanie, że będzie można je produkować i przede wszystkim eksportować jeszcze w 1995 roku. Na dodatek wszystkie wymienione w programie komputery powinny już być wedle harmonogramu wdrożone do produkcji seryjnej. Tymczasem część jeszcze nie ujrzała światła dziennego, natomiast przyszłość podstawowego komputera edukacyjnego — "Elwro 800 Junior" jest w momencie, gdy piszę te słowa wciąż mocno niepewna — ruszyła produkcja, lecz jednocześnie utknęła sprzedaż. Od listopada 1987 do połowy stycznia br. Ministerstwo Edukacji Narodowej nie wykupiło już gotowych komputerów. Wygląda zatem, że początki realizacji programu są bolesne. Oby nie okazał się jeszcze jedną papierową wizją.

Oczywiście produkcja jednostek centralnych to jeszcze nie wszystko. Potrzeba monitorów, klawiatur, drukarek, stacji dysków, winchesterów, ploterów itd. Jak na razie w ofercie przemysłu państwowego pustka. Są już natomiast liczne Centralne Programy Badawczo-Rozwojowe w wyniku których powstać mają prototypy nowych urządzeń. Ich wdrożenie nie będzie jednak tak szybkie, jakby się chciało — np. na twarde dyski rodzimej produkcji przyjdzie nam czekać do 1991 roku. Najwyraźniej jednak nie liczymy na żadną długofalową współpracę, czy też kooperację, zarówno ze Wschodem, jak i z Zachodem. Program bowiem przewiduje niemal własny model rodzimego przemysłu komputerowego. Jeśli coś nawet importujemy, to tylko przejściowo. Docelowo produkować mamy wszystko. Pytanie tylko: po co? Czy nie lepiej byłoby spróbować kooperacji, tym bardziej, że np. Węgrów, czy Czechosłowaków też z pewnością nie stać na samodzielne rozwiązanie wszystkich problemów technicznych i konstrukcyjnych w zakresie hardware'u.

Znaków zapytania jest zresztą więcej. Dotyczą one m.in. rozwoju sieci teleinformatycznych — program mówi o tym dość enigmatycznie, czy wreszcie, co bardziej nas chyba jeszcze interesuje, produkcji komputerów domowych. Z zapisów programu wynika, że wciąż jeszcze głównym źródłem zaopatrzenia prywatnych hobbystów w komputery pozostanie "Pewex" i import prywatny. Zakłada się jedynie wprowadzenie na rynek nadwyżki mikrokompterów edukacyjnych. Czyżby autorzy nie wiedzieli, że jak na razie na świecie tylko nieliczne konstrukcje sprawdzają się w szkole równie dobrze jak w domu?

Podczas posiedzenia KNiPT zatwierdzającego program uzasadnione kontrowersje wzbudziło pominięcie de facto w programie całego ruchu klubów komputerowych — i tego związanego z organizacjami młodzieżowymi, i "niezorganizowanego". Zwrócił na to uwagę sekretarz generalny Rady Krajowej TMMT Wojciech Wyszomirski. Na jego wniosek opracowany ma być stosowny rozdział programu poświęcony właśnie społecznemu ruchowi komputerowemu.

Wszystkie plany, programy i zamierzenia dotyczące polskiej komputeryzacji mają jednak wielkie "ale". Możemy produkować mikrokomputery, drukarki, plottery, stacje dysków, ale potrzebne są podzespoły. Dopóki nie podejmie się dramatycznego wręcz wysiłku dla modernizacji CEMI i pozostałych zakładów podzespołowych i nie rozpocznie własnej produkcji choćby części potrzebnych "kości" parafrazując Sienkiewicza powiedzieć można: "Szkoda howoryty, szkoda pisaty". A na podjęcie działań rzeczywiście nadzwyczajnych jakoś się nie zanosi. Sumaryczne wielkości nakładów na przemysł komputerowy w latach 1987-1995 będą porównywalne z kosztami budowy jednej kopalni. A to mimo wszystkich deklaracji "Programu..." stawia go raczej w sferze życzeń. A zatem wszystkiego najlepszego polska komputeryzacjo!

Grzegorz Onichimowski

O ZMIENNYCH DYNAMICZNYCH SŁÓW KILKA

Zwolennikom i przeciwnikom Pascal-a z pewnością obrzydło już nieustanne wyliczanie zalet tego języka na łamach pism komputerowych (Znacie? No to posłuchajcie: modularność. przejrzysta struktura programów. definiowanie typów danych, zasada predefinicji pojeć, możliwość użycia rekurencji...)

Autorzy hymnów na cześć Pascal-a rzadko kiedy dostrzegają jednak pewna inną cechę tego języka, jaka jest możliwość użycia zmiennych dynamicznych. Rzecz to niebanalna, a całkiem obca nawet najwspanialszym wersjom Basic-a czy Fortranu..

O CO CHODZI?

To znaczy — co to są owe zmienne dynamiczne? Otóż wiemy, że w Pascal-u mamy możliwość deklarowania dwóch rodzajów zmiennych — globalnych i lokalnych. Zmienne takie istnieją tylko w trakcie wykonania części programu, w której zostały zadeklarowane. Oznacza to, że w momencie, gdy rozpoczyna się wykonywanie np. procedury, program rezerwuje sobie odpowiedni obszar w pamięci dla przechowywania jej zmiennych lokalnych. Wraz z zakończeniem wykonywania procedury obszar ten jest zwalniany i może być wykorzystany do innych celów. Oczywiście tracimy przy tym wartościowych zmiennych.

W odniesieniu do takich zmiennych ukuto nazwę "statyczne", ponieważ istnieją one w pamięci przez cały czas wykonania tego modułu, w którym je zadeklarowaliśmy. Być może — rozumujemy więc — zmienne dynamiczne to takie zmienne, które możemy tworzyć i usuwać według własnego uznania, w dowolnym momencie trwania programu? Tak, o to właśnie chodzi.

Ale po co?

Oto przykład.

Wyobraźmy sobie zapisany na dysku plik, który zawiera pewne rekordy. Jak wczytać jego zawartość do pamięci? Gdybyśmy dysponowali tylko zmiennymi statycznymi, należałoby umieścić kolejne wczytywane rekordy w tablicy. No tak, ale każdy z łatwością dostrzeże tu dwie kardynalne wady:

1. Rozmiar tablicy określa maksymalną liczbę rekordów, jaka może być wczytana. Plik natomiast może zawierać więcej rekordów;

2. Z drugiej strony, plik może zawierać bardzo mało (nawet zero) rekordów. W takim wypadku duża część naszej tablicy jest bezużyteczna. Nie ma strachu, jeżeli nasz program używa tylko jednej takiej tablicy. Co jednak zrobić, kiedy chcemy jednocześnie operować kilkoma takimi plikami i pomieścić je w pamięci jednocześnie? Na zadeklarowanie kilku

wystarczających rozmiarami tablic musielibyśmy poświęcić ogromne ilości pamięci.

Widzimy tu wyraźnie sens użycia zmiennych o rozmiarach, które moglibyśmy regulować w miarę naszych bieżących potrzeb. Zastosowań dla zmiennych dynamicznych można znaleźć wszakże daleko więcej — stwarza je praktyka.

Wiemy już zatem o co chodzi, pozostaje jeszcze pytanie

JAK?

Kluczem do zrozumienia mechanizmu posługiwania się zmiennymi dynamicznymi jest pojęcie (znów termin ukuty dość dawno) wskaźnika. Wyobraźmy sobie następujące typy danych:

```
TYPE NaszRekord = RECORD
                    Wys, Szer, Dług: INTEGER:
                    Masa: REAL:
                    END;
     Wskaźnik  = ↑ NaszRekord:
VAR P: Wskaźnik;
```

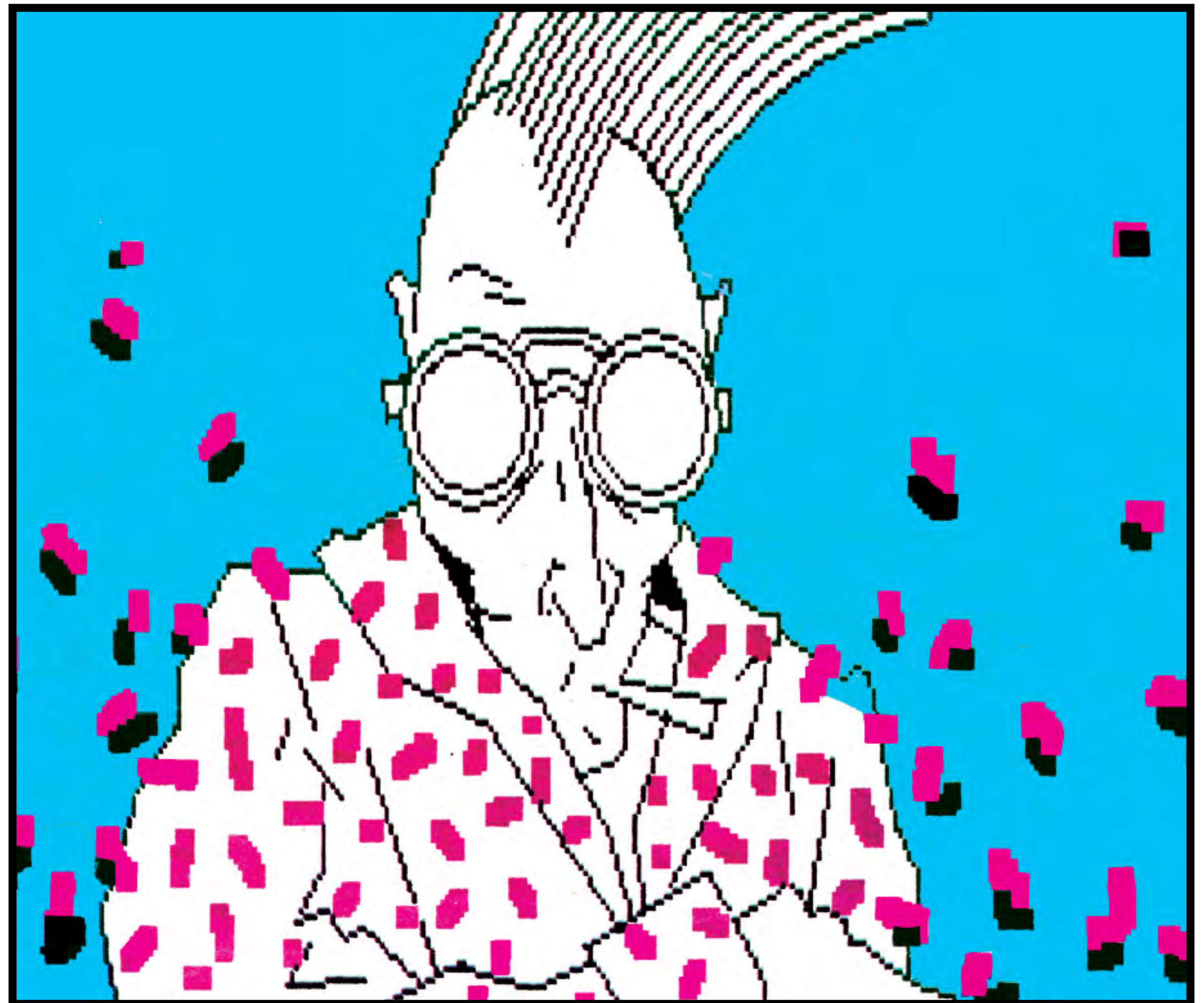
Definicja typu NaszRekord jest jasna. Wątpliwości budzi definicja typu Wskaźnik (ta podejrzana strzałka...). Czym jest zmienna P, należąca do tego typu?

Strzałka w definicji typu Wskaźnik oznacza, że zmienne tego typu mogą wskazywać na obiekty typu NaszRekord — wymienionego zaraz za strzałką. Co to znaczy "wskazywać"? To znaczy, że wartością takiej zmiennej może być adres (nie wnikamy w to, jak jest on reprezentowany) pewnego, znajdującego się w pamięci rekordu typu NaszRekord. Możemy tę sytuację przedstawić sobie na rysunku (rys. 1).

Tenże wskaźnik stanowi właśnie narzędzie do posługiwania się zmiennymi dynamicznymi. Jak się to robi? To proste: oto podstawowe operacje, jakich możemy dokonywać na wskaźnikach.

1. Możemy je porównywać, ale tylko przy pomocy operacji = i o. Równość oznacza, że wskaźniki wskazują na ten sam obiekt, nierówność — że na różne.

2. Musimy mieć jakąś możność wyrażenia faktu, że wskaźnik nie wskazuje na żaden obiekt znajdujący się w pamięci. W tym celu Pascal zawiera stałą standardową NIL. Oznacza ona po prostu "adres nieistniejący".



Oto przykłady użycia:
P := **NIL**;
IF Q = **NIL THEN**...

3. Na zmienną będącą wskaźnikiem możemy podstawiać NIL lub wartość dowolnej innej zmiennej — byle należała ona do tego samego typu wskaźnikowego (wskaźnik wskaźnikowi nierówny — zależnie od tego, na jakiego typu obiekt wskazuje).

P := Q; Q:= **NIL**;

4. Ważna operacja:

NEW(P);

jest standardową procedurą Pascal-a. Jej działanie jest następujące: zostaje utworzony w pamięci nowy obiekt typu wskazywanego przez parametr, a następnie parametr zaczyna wskazywać na ów obiekt. W powyższym przykładzie w pamięci zostanie zarezerwowane miejsce na nowy rekord typu NaszRekord, zaś zmienna P zaczyna to miejsce — inaczej: nowo utworzony rekord — wskazywać.

5. No tak, ale ja teraz dostać się do poszczególnych pól nowego rekordu? Skoro P nam go wskazuje, pozostaje tylko użyć następujących oznaczeń:

P ↑ — oznacza obiekt wskazywany przez P, a zatem cały rekord typu NaszRekord.

P .Masa — oto, jak dostajemy się do poszczególnych pól wskazywanego rekordu. Gdyby obiektem wskazywanym była np. tablica, to prawidłowy mógłby być np. zapis P f [10]. Oto kilka przykładów:

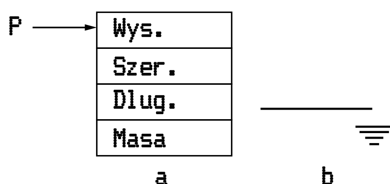
```
P↑ -Masa := 1.05;
WITH P↑ DO BEGIN
  Wys := 100;
  Szer := 10;
END;
P ↑ Dług := P ↑ .Szer;
```

6. Musimy jeszcze mieć możność usunięcia wybranych struktur z pamięci. Realizację takiego żądania zapewnia standardowa procedura DISPOSE. Jej parametrem jest znów wskaźnik.

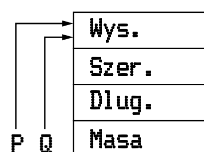
DISPOSE (P);

Procedura ta powoduje, że zostaje usunięty z pamięci obiekt wskazywany przez P (to znaczy, że zajmowana przezeń pamięć jest zwalniana).

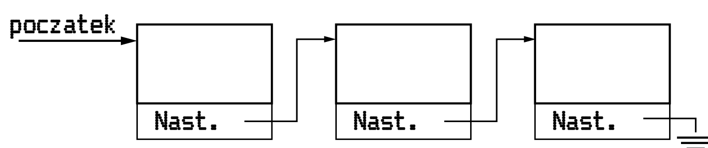
Uzbrojeni w ten zestaw operacji możemy zabrać się do pracy. Na początek zastanówmy się nad problemem przechowywania w pamięci ciągu rekordów, który postawiliśmy na początku.



rys. 1



rys. 2



rys. 3

STRUKTURY LISTOWE

Jak zatem reprezentować w pamięci taki ciąg rekordów, postępując w sposób oszczędny?

Sęk w tym, aby zbudowana struktura danych zajmowała w pamięci tylko tyle miejsca, ile potrzebne jest dla wczytanych rekordów — nie więcej.

Zaproponujemy tu rozwiązanie, które w praktyce okazuje się najbardziej ekonomiczne — mianowicie uszeregowanie naszych rekordów w listę. Czym jest lista? To proste: zauważmy, że wewnątrz rekordu NaszRekord możemy umieścić pola dowolnych typów. Zadeklarujmy tam więc dodatkowo pole typu Wskaźnik. Wtedy rekord typu NaszRekord będzie mógł sam zawierać wskaźnik do innego rekordu, ten zaś do innego, ten do innego... i tak dalej, aż w końcu wskaźnik w którymś rekordzie byłby równy NIL — i na tym lista by się zakończyła. Dla lepszego wyobrażenia sobie takiej sytuacji mamy rysunek 2. Pokazuje on dodatkowo, że zachodzi potrzeba posiadania wskaźnika do pierwszego rekordu takiego ciągu. W przeciwnym wypadku nie mielibyśmy dostępu ani do niego, ani do żadnego z pozostałych rekordów listy.

Spróbujmy zatem zdefiniować odpowiednie typy danych.

```
TYPE Wskaźnik = ↑NaszRekord;
   NaszRekord  = RECORD
                 Wys.Szer.Dlug : INTEGER;
                 Masa : REAL;
                 Nast : Wskaźnik
                 END;
VAR Początek, P, Q: Wskaźnik;
```

Zaraz, zaraz — powie zaraz spostrzegawczy czytelnik — w tej definicji typów nie jest spełniona zada predefinicji, bo typ NaszRekord jest użyty ZANIM w tekście wystąpi jego definicja. Czytelnik będzie miał rację. więcej, w tym konkretnym przypadku

nie jest możliwy taki zapis powyższego fragmentu programu, aby zasada definiowania pojęć przed ich użyciem była zachowana w pełni. Dlatego też mamy tu do czynienia z JEDYNYM dopuszczonym w Pascal-u odstępstwem od zasady predefinicji. Mówiąc ściślej, typ wskaźnikowy wolno w tym języku zdefiniować zanim zdefiniuje się typ obiektu wskazywanego.

Skoro już zdefiniowaliśmy typ takiej struktury — jak ją samą utworzyć?

Potrzebna będzie nam jedna zmienna typu Wskaźnik, która pokazywałaby stale na pierwszy z ciągu rekordów. Do wykonywania operacji na rekordach dalszych możemy użyć pomocniczych zmiennych wskaźnikowych, które wskazywałyby na rekordy aktualnie przez nas przetwarzane. Można też postąpić inaczej i w celu uzyskania dostępu do rekordów wewnątrz listy posłużyć się algorytmami rekurencyjnymi. Program przykładowy zamieszczony obok pokazuje, że ich użycie aż się prosi w tym przypadku.

Program ten stanowi rozwiązanie postawionego na początku artykułu problemu na przykładzie prostego notesu telefonicznego. Do wykonywania wszelkich operacji na liście rekordów użyto procedur rekurencyjnych — warto je przeanalizować. Podobne algorytmy, zapisane nierekurencyjnie, znaleźć można w książce N. Wirtha "Wstęp do programowania systematycznego".

Na zakończenie kilka praktycznych uwag, dotyczących posługiwania się zmiennymi wskaźnikowymi.

1. Trzeba pamiętać, że wskaźnikom nie jest nadawana żadna inicjalna wartość. W szczególności należy pamiętać, aby wskaźnikom nadawać na początku programu wartość NIL. Odwołanie poprzez wskaźnik o nieokreślonej wartości prowadzić może do trudnych do wykrycia błędów.

2. Należy pamiętać o tym, by nie dopuścić do istnienia obiektów dynamicznych nie wskazywanych przez żaden wskaźnik. Do takich obiektów nie sposób się dostać; również nie można ich usunąć procedurą DISPOSE.

3. Jeżeli tworzymy nowy obiekt wskazywany przez NEW (P), to przed następnym wywołaniem NEW (P) musimy zadbać o zachowanie wartości P, które jest wtedy wskaźnikiem do pierwszego utworzonego obiektu. W przeciwnym przypadku ów obiekt znajdzie się w sytuacji z punktu 2.

4. Trzeba uważać, by nie odwoływać się do obiektu wskazywanego przez zmienną, gdy jest ona równa NIL. W wielu przypadkach narzucałoby się np. użycie takiej instrukcji:

IF** (P<-**NIL) AND (**P↑ .Masa = O) **THEN...

...nie jest to jednak prawidłowe. Pascal wartościuje bowiem wszystkie człony wyrażenia logicznego i z tego powodu będzie starał się obliczyć Pf .Masa niezależnie od tego, czy P jest równe NIL, czy też nie. Prawidło- wo należy tę instrukcję rozbić na dwie:

IFP<>NIL THEN

IF** P↑ .Masa = O **THEN...

5. Należy uświadomić sobie, że instrukcja przypisania w odniesieniu do wskaźników ma nieco inne skutki, niż do tego przywykliśmy. Normalnie bowiem efektem przypisania

Q:= P;

jest skopiowanie wartości zmiennej P do zmiennej O. W przypadku, gdy P i Q są zmiennymi wskaźnikowymi dzieje się podobnie, ale kopiowane są wskaźniki, a nie obiekty wskazywane! Jeżeli np. P wskazuje pewien rekord, to efektem powyższego przypisania będzie sytuacja jak na rys. 3.

W tej sytuacji instrukcja np.

P .Masa : = O

zmienia jednocześnie wartość pola Q .Masa — bo przecież w gruncie rzeczy są to te same pola! Jeszcze gorzej jest w przypadku, gdy wykonamy DISPOSE (O). Wskaźnik P nie zmienia przy tym oczywiście swojej zawartości, ale dalsze odwoływanie się do wskazywanego przezeń rekordu nie ma sensu i może prowadzić do przykrych błędów. Dlatego też w przypadku usuwania obiektu wskazywanego przez wiele wskaźników należy je wszystkie wyszukać i nadać im wartość NIL.

Sądzę, że publikowany program stanowi przykład rzeczywistej wygody, jaką daje posługiwanie się listami. Dla porównania proponuję przejrzeć artykuł "Naprawdę nie święci garnki lepią" z nru 3/87 "Bajtka". Opisane tam proste bazy danych spełniają funkcje podobne do naszej — ale dla uzyskania podobnego efektu w Basic-u trzeba było uciekać się do niewygodnych metod typu przechowywania rekordów w tablicy — a o wadach tego rozwiązania już wiemy.

Program został napisany w Turbo Pascal-u. Ponieważ opis tego języka był już w " Bajtku" publikowany, sądzę, że w razie potrzeby da się on bez trudu przełożyć na inny dialekt Pascal-a.

Marek Wyrwidąb

LITERATURA

1. **Nikiaus Wirth "Wstęp do programowania systematycznego", wyd. I, WNT 1982 — seria BIO;**
2. **M. Iglewski, J. Madey, S. Matwin "Pascal", wyd. III, WNT 1984 — seria BIO.**



```
PROGRAM NotesTelefoniczny;
(*************************************)
(* Implementacja podstawowych operacji *)
(* na liscie, na przykladzie procedur  *)
(* obslugi prostej bazy danych         *)
(* Przykład uzycia algorytmow rekuren- *)
(* cyjnych do obslugi list.            *)
(*        B A J T E K    1988          *)
(*************************************)
TYPE
 Slowo = strng [20];
 Numer = string [71;
 Wsk   = ^Zapis
 Zapis = record
          Nazwisko:Slowo;
          Telefon :Numer;
          Nast    :Wsk;
         end;
VAR
 PoczNotesu:Wsk;
 F         :text;
 Lancuch   :Slowo;
 Menu,
 Koniec    :boolean;
 Zn        :char;
 Nazw      :Slowo;
 Nowy      :Wsk;


PROCEDUERE Wyswietl (P;Wsk);
begin  (* Wypisywanie zawartosci listy *)
  if P = nil then
   writeln('Nie ma (wiecej) rekordow.')
  else begin
  write(P^.Nazwisko);
  write(' ':20-length (P^,Nazwisko));
  writeln (P^.Telefon: 7);
  Wyswietl (P^.Nast)
 end;
end;

PROCEDUDE Szukaj (P:Wsk; Kogo:Slowo);
begin (* Wyszukanie wskazanego rekordu *)
 if P = nil then
  writeln ('Poszukiwanie bez rezultatu.')
 else
  if P^.Nazwisko = Kogo then
   writeln (P^.Telefon:10)
  else
   Szukaj (P^.Nast, Kogo);
 end;

PROCEDURE Wstaw (VAR P, Nowy:Wsk);
begin (* Wstawienie nowego rekordu *)
      (* w porzadku alfabetycznym  *)
 if P = nil then begin
  Nowy^.Nast := P;
  P := Nowy
 end
else
 if P^.Nazwisko < Nowy^.Nazwisko then
   Wstaw (P^.Nast, Nowy)
 else begin   Nowy^.Nast := P;
  P := Nowy
 end end;

PROCEDURE NowyZapis (VAR Nowy:Wsk); begin
 new (Nowy);
 write ('Nazwisko ? ');
 read (Nowy^.Nazwisko); writeln;
 write ('Telefon ? ');  read (Nowy^.
Telefon); writeln;
end;

PROCEDURE Usun (VAR P:Wsk; Kogo:Slowo);
VAR Pom:Wsk;
begin (* Usuniecie wskazanego rekordu *);
 if P = nil then
  writeln ('Brak szukanego rekordu.')
else
if P^.Nazwisko = Kogo then begin
 Pom := P;
 P := P^.Nast;
 dispose (Pom)
end else
 Usun (P^.Nast, Kogo)
end;

PROCEDURE Kasuj (VAR P:wsk);
begin (* Usuniecie calej listy *)
 if P <> nil then begin
  Kasuj (P^.Nast);
  dispse(P);
  P := nil end
end;

PROCEDURE Pisz (VAR P:Wsk; VAR F:tant);
begin (* Zapisanie listy na dysku *)
 if P = nil then
  Close(F)
 else begin
  writeln (F, P^.Nazwisko);
  writeln (F, P^.Telefon);
  Pisz (P^.Nast, F)
 end
end;

PROCEDURE Czyt (VAR P:Wsk, VAR File)
begin  (* Odczyt listy z dysku *)
 if eof (F) then begin
  P := nil;
  Close (F)
 end
 else begin
  new (P);
  readln (F, P^.Nazwisko);   readln (F,
P^.Telefon);   Czyt (P^.Nast, F)
 end
end;

FUNCTION Potwierdzone:boolean;
var Zn:char;
begin
 write ('Na pewno ? (T/N) ');
 read  (Kbd, Zn); writeln (Zn);
 case Zn of
   'T','t' : Potwierdzone = true
 else Potwierdzone := false
 end
end;

begin (* Program udcstepniajacy funkcje *)
      (* obslugi listy                  *)
 PoczNotesu := nil;
 Koniec     := false;
 Menu       := true;
 Assign (F, 'NONAME.TEL')
 while not Koniec do begin
  writeln;
  if Menu then begin
   writeln ('-=-=-=-=-M-E-N-U-=-=-=-');
   writeln ('<Spacja> - wyswietlenie');
   writeln ('S        - wyszukiwanie');
   writeln ('W        - wstawianie  ');
   writeln ('U        - usuwanie    ');
   writeln ('Z        - zapis       ');
   writeln ('O        - odczyt      ');
   writeln ('K        - kasowanie   ');
   Menu := false;
  end;
  write ('Co Kazesz ? ');
  read (Kbd, Zn); writeln (Zn);
  case Zn of
   ' '     :begin
             write   ('NAZWISKO');
             writeln  ('TELEFON':19);
             writeln;
             Wyswietl (PoczNotesu)
            end;
   'S','s':begin
            write ('Nazwisko ? ');
            read (Nazw); writeln;
            Szukaj (PoczNotesu, Nazw)
           end;
   'W",'w' :begin
            NowyZapis (Nowy);
            Wstaw (PoczNotesu, Nowy)
           end;
   'U','u' :begin
            write ('Nazwisko ? ');
            read (Nazw); writeln;
if Potwierdzone then
            Usun (PoczNotesu, Nazw)
end;
   'K','k':if Potwierdzone then
           Kasuj (PoczNotesu);
   'Q','q':if Potwierdzone then
           Koniec := true;
   'O','o':if Potwierdzone then begin
           Kasuj (PoczNotesu);
           reset (F);
           Czyt (PoczNctesu, F)
          end;
'Z','z':begin
         rewrite (F);
         Pisz (PoczNotesu, F)
        end
else Menu := true
 end; end;
writeln "Czesc !'");
end.
```

PORTY JOYSTICKÓW (1)

Wspominaliśmy już w "Bajtku" o możliwości wykorzystania portów joysticków jako wyjść do sterowania urządzeń zewnętrznych, Teraz pora na rozszerzenie tego zagadnienia.

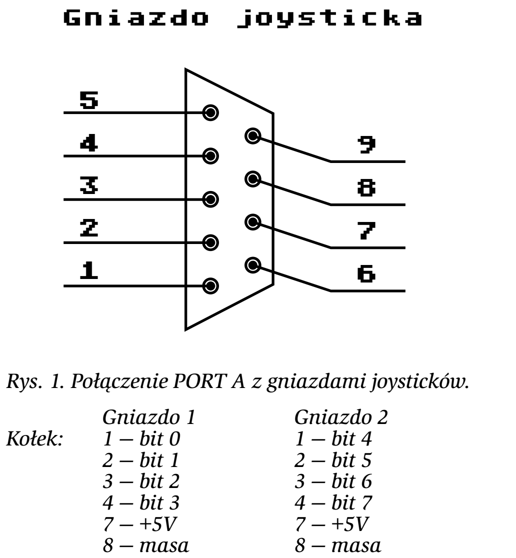
W komputerach Atari gniazda joysticków połączone są z układem o nazwie PIA, którym jest standardowy programowalny układ wejścia/wyjścia 6520. Składa się on z dwóch programowalnych portów 8-bitowych oraz dwóch wejść kontrolnych dla każdego portu. Dzięki temu, że oba porty w układzie 6520 są programowalne istnieje możliwość zaprogramowania każdej linii sygnałowej jako wejścia lub wyjścia.

Na początku omówimy możliwości sterowania urządzeń zewnętrznych przez pony joysticków, przy czym interesować nas będzie jedynie port A układu 6520 (port B wykorzystywany jest do zarządzania bankami pamięci i kontroli złącza szeregowego).

Wstępny opis układu PIA oraz przypisanie poszczególnych bitów portu A układu 6520 do styków przedstawione zostało w "Bajtku" 10/87.

Kontrolę nad portem A uzyskuje się poprzez rejestry PORTA (54016, $D300) oraz PORTACNTL (54018, SD302). Wpisując odpowiednie wartości do rejestru PORTACNTL uzyskuje się różne warianty pracy układu 6520.

Znaczenie poszczególnych bitów w tym rejestrze jest następujące:



Rys. 1. Połączenie PORT A z gniazdami joysticków.

	Gniazdo 1	Gniazdo 2
Kołek:	1 – bit 0	1 – bit 4
	2 – bit 1	2 – bit 5
	3 – bit 2	3 – bit 6
	4 – bit 3	4 – bit 7
	7 – +5V	7 – +5V
	8 – masa	8 – masa

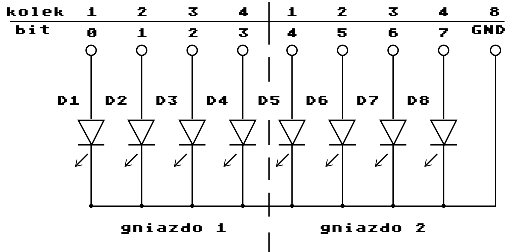
UWAGA: Nie należy wpisywać liter wydrukowanych przed numerem linii programu. NIE SĄ one częścią linii, lecz kodem kontrolnym "Edytora BASIC a" (zob. "Bajtek" 1/88 lub "Bajtek-Atari").

Bit 0 i 1 — sterowanie wejściem kontroli CA1
Bit 2 — sterowanie portem A
Bit 3 — 5 — sterowanie wejściem kontroli CA2
Bit 6 — zgłoszenie przerwania maskowalnego IRQA1 pochodzące z wejścia CA2
Bit 7 — zgłoszenie przerwania maskowalnego IRQA2 pochodzące z wejścia CA1

W tym artykule nie będziemy się zajmować znaczeniem bitów 0, 1, 3-7, interesować nas będzie jedynie bit 2. Jeżeli będzie on równy 0 (mówimy wtedy, że bit ten jest skasowany), to PORTA będzie pracował jako rejestr porządkowania danych. Co to oznacza? Otóż każdemu bitowi tego reiestru odpowiada jedna linia doprowadzona do gniazda joysticka. Jeżeli jakiś bit w tym rejestrze będzie skasowany, to odpowiadająca mu linia w gnieździe joysticka będzie spełniać rolę wejścia. Jeżeli natomiast ustawimy ten bit (zmienimy jego wartość z 0 na 1), to odpowiednia linia w gnieździe joysticka spełniała będzie rolę wyjścia. Gdy bit 2 w PORTACNTL jest ustawiony (1), to dane z gniazda joysticka są przesyłane do PORTA (przy ustawieniu jako wejście) lub z PORTA do gniazda joysticka (przy ustawieniu jako wyjście).

Tyle wstępu, przejdźmy teraz do praktycznego wykorzystania przedstawionych informacji.

Należy przy tym przestrzec przed niestarannym lub niedokładnym wykonywaniem opisanych tu rozwiązań, NIEDOPUSZCZALNE SĄ PRZYPADKOWE ZWARCIA, gdyż może to spowodować uszkodzenie układu 6520. Mówiąc prościej: chwila nieuwagi lub niestaranny montaż i pozbywamy się komputera na kilkanaście dni tracąc przy tym kilka tysięcy złotych na wymianę układu 6520, Nie polecamy więc samodzielnych prób osobom, które nie mają doświadczenia w montażu układów elektronicznych.



Rys. 2. Przyłączenie diod do gniazd joysticków (D1-D8 — dowolne diody elektroluminescencyjne typu CQYP)

Do podłączenia układów do komputera najlepiej używać typowych kabli z wtyczkami z uszkodzonych joysticków. Można też wykorzystać polskie złącza tzw. komputerowe nr. 871 (ELTRA).

Do gniazda 1 joysticka podłączone są 4 mniej znaczące bity portu A (tzn. bity 0, 1, 2 i 3), pozostałe (4, 5, 6 i 7) do gniazda 2 joysticka. Połączenia te przedstawione są na rys. 1.

Pierwszy układ próbny jest przedstawiony na rys. 2. Spróbujmy go zmontować i podłączyć do portów joysticków.

Niektórzy mogą być zdziwieni brakiem rezystorów szeregowych w układzie, ale gniazda joysticków połączone są wewnątrz, komputera z układem 6520 przez rezystory o wartości 2,2 kOhma. Rezystory te zabezpieczają układ 6520 przed przeciążeniem prądowym, ich duża wartość powoduje, że diody będą świecić bardzo słabym światłem. Znajdzie się na to rada, ale o tym nieco później.

Po podłączeniu układu włączamy komputer i wpisujemy poniższy program.

```
QG 10 GRAPHICS  18
CV 20 P=PEEK(54018):POKE 54018,P-4:REM PORT
   A JAKO REJESTR PORZADKOWANIA DANYCH
IF 30 POKE 54016,255:REM WSZYSTKIE BITY
   PORTU A USTAWIONE JAKO WYJSCIA
PV 40 POKE 54018,P:REM ODTWORZENIE
   POPRZEDNIEJ WARTOSCI PORTACNTL
ML 50 FOR I=1 TO 255:POKE 54016,I
QQ 60 POSITION 7,61:? #6;I;"  "
PV 70 FOR K=0 TO 500:NEXT K:REM PETLA
   OPOZNIAJACA
IW 80 NEXT I
VK 90 IF PEEK(53279)<>6 THEN 50:REM KONIEC
   PO NACISNIECIU START
DP 100 P=PEEK(54018):POKE 54018,P-4
YA 110 POKE 54016,0:REM WSZYSTKIE BITY
   PORTU A USTAWIONE PONOWNIE JAKO WEJSCIA
KD 120 POKE 54018,P
NW 130 END
```

Po uruchomieniu programu zaobserwujemy zapalanie się diod w różnej kolejności. Jeżeli diody będą ustawione w takiej kolejności, jak na rys. 2, to obserwować będziemy binarne (dwójkowe) przedstawienie liczby wyświetlanej na ekranie, przy czym dioda zapalona odpowiada logicznej jedynce, a dioda zgaszona logicznemu zeru.

W następnych częściach artykułu przedstawione zostaną rozszerzenia powyższego układu oraz układy bardziej skomplikowane i poważniejsze o praktycznym zastosowaniach.

Przemysław Strzelecki
Wojciech Zientara

TAJEMNICE ATARI (6)

Jak na razie dotarł tylko jeden list od czytelników Bajtka. w którym opisana byłaby sytuacja pomagająca wygrać w grze jej uczestnikowi.

Jego autorem jest Maciej Gradowski z Krakowa. Czyżby użytkownicy komputerów ATARI nie byli w stanie odkryć ich sekretów? W to nie wierzymy i w dalszym ciągu czekamy na listy z "tajemnicami".

Tymczasem podajemy kody niezbędne do teleportacji w najnowszym przeboju komputerowej giełdy — grze "Star Quake". Są to: Delta, Triad, Penta, Kernx, Atari, Whole. Salco, Artic, Minim. Argon, Cosec, Crash, Secon, Z.A.P. i Quark.

W grze "Hacker" na pytanie komputera o kod wywoławczy (LOGON PLEASE) odpowiadamy wpisując za pomocą klawiatury słowo **Australia**. Wykonanie takiego posunięcia uwalnia gracza od przechodzenia przez test robota SRU i pozwala od razu przejść do części zasadniczej gry. Na tym etapie uczestnik zabawy ma cztery szanse, aby zostać złapanym przez obcego satelitę bez ponoszenia konsekwencji tego faktu. Warunkiem determinującym dalszą kontynuację gry jest znajomość kodów rozpoznawczych. Oto one:

— pierwszy raz MAGMA, LTD
— drugi raz AX-0310479
— trzeci raz HYDRAULIC
— czwarty raz AUSTRALIA
— piąty raz KONIEC GRY

Dla tych, którzy nie chcą ułatwiać sobie życia i omijać tekstu robota podajemy elementy niezbędne do identyfikacji. Są to:

— INFRARED VIDEO IMAGE SENSOR (ośrodek centralno-wizyjny robota)
— ASYNCHRONUS DATA COMPACTOR (główny receptor danych dotykowych)
— HYDRAULIC MOTIVATOR (główny ośrodek napędowy robota)
— PHLAMSON JOINT (plazmowy amortyzator kierunkowy)
— THELMAN PORT (złącze urządzeń zewnętrznych).

"Montezuma's Revenge" jest dla każdego posiadacza Atari czymś w rodzaju standardu. Z reguły od tej gry zaczyna się wspólną zabawę z komputerem. Być może zaskoczymy niektórych, ale jest sposób na to, aby utrudnić przejście Pedra przez tak dobrze znane już komnaty grobowca faraonów. Wystarczy tylko po załadowaniu programu do pamięci nacisnąć klawisz RESET, a następnie klawisz SELECT. Dzięki temu potroi się liczba śmiercionośnych czaszek na aktualnej planszy gry i dotarcie do skarbów Montezumy nie będzie już dziecinną igraszką dla najbardziej nawet zaawansowanych zawodników.

Ciekawe, czy wiecie jak zwolnić tempo walki w "Karate International". Rozwiązanie tej zagadki tkwi w klawiszu X. Wystarczy go nacisnąć a następnie wybrać numer od jednego do czterech, pamiętając o zasadzie, że im wyższa cyfra tym wolniejsze ruchy wykonuje kierowana przez joystick postać na ekranie. Dodatkowo w tej wersji gry istnieje możliwość zmiany aktualnej planszy gry. Wykonuje się to za pomocą klawisza OPTION.

W grze pod tytułem "Quasimodo" o ile chce się przejść do dalszych etapów należy w pierwszej komnacie strącić co najmniej trzech łuczników wspinających się po drabinach, ale w taki sposób aby żaden następny nie był widoczny. Aby tego dokonać trzeba użyć kul ułożonych w stosy na szczycie muru.

Na ten raz to już wszystko. Dziękujemy "pierwszemu odważnemu" Maćkowi, że do nas napisał i czekamy na jeszcze.

Sergiusz Piotrowski
Tomasz Mazur

MIGAJĄCY KURSOR

Po zamieszczeniu w ostatnim odcinku cyklu "Nie bój się przerwań" prośby o nadsyłanie własnych procedur otrzymaliśmy od Czytelników kilkanaście listów. Dziwnym trafem wielu użytkownikom Atari przeszkadza brak migotania kursora. Wybraliśmy trzy programy, które usuwają tę niedogodność (?). Dla wszystkich, którzy chcieliby bliżej poznać przerwania Atari, zamieszczamy oprócz wydruków w Basic-u także wydruki programów źródłowych w asemblerze MAC/65. Zostały one odtworzone w redakcji, gdyż autorzy "zapomnieli" je przysłać. Dwa pierwsze programy wykorzystują przerwanie zegara TIMER 2, jednak w różny sposób. W pierwszym migotanie znaku jest realizowane przez ingerencję w rejestr kontroli wyglądu znaków ekranowych (CHARCNT), drugi natomiast ingeruje w pamięć obrazu przy użyciu rejestru zawierającego adres kursora. Ostatni program wykorzystuje przerwanie synchronizacji pionowej i rejestr zegara systemu. Dodatkowo podczas procedury wywoływanej przez RESET ustawiane są różne parametry komputera. Dzięki temu program ten może być szczególnie przydatny dla osób dużo pracujących z komputerem.

(redakcja)

Jedną z wad edytora BASIC-a w komputerze Atari 800XL jest brak migającego kursora, co utrudnia jego lokalizację. Aby to poprawić ułożyłem krótki program.

Po jego uruchomieniu kursor zaczyna migać. Prędkość migania można zmienić poprzez zmianę zawartości komórki 1549 (zalecana 20). Kursor przestaje migać po naciśnięciu RESET; aby przywrócić stan poprzedni należy wykonać sekwencję rozkazów z linii 90—120. Kursor nie przestaje migać po zmianie trybu graficznego. Procedura jest relokowalna; adres początku umieszczany jest w komórkach 552 i 553.

Uwaga: przy dużej prędkości wyprowadzania danych na ekran (np. listowanie) kursor może zostawiać ślad.

Bartosz Polednia

```
10 REM *************************
20 REM *    MIGAJACY KURSOR    *
30 REM *  (C) 1987 B.POLEDNIA
*
40 REM *************************
50 RESTORE
60 FOR A=1536 TO 1553
70 READ X:POKE A,X
80 NEXT A
90 POKE 552,0:POKE 553,6
100 POKE 538,10:POKE 539,0
110 POKE 54286,64
120 DATA 162,0,161,94,73,128
130 DATA 129,94,73,0,129,94
140 DATA 169,20,141,26,2,96
```

```
0100 ;Migajacy kursor - MAC/65
0110 ;
0120 OLDADR = $5E
0130 TIMCOUNT2 = $021A
0140 *=$0600
0150 ;
0160  LDX #$00
0170  LDA (OLDADR,X)
0180  EOR #$80
0190  STA (OLDADR,X)
0200  EOR #$00
0210  STA (OLDADR,X)
0220  LDA #$14
0230  STA TIMCOUNT2
0240  RTS
```

Program napisany w języku maszynowym wykorzystuje przerwanie VBLK. Na uwagę zasługuje fakt, że program działa dalej po naciśnięciu RESET. Jeśli chcemy wyłączyć pulsowanie kursora, to wystarczy POKE 9,0 i RESET. Powrót do migania uzyskamy przez POKE 9,1 i RESET. Program umieszczony jest na szóstej stronie pamięci.

Adam Hartwig

```
0 REM MIG-CURSOR
10 REM (C)1987 ADAM HARTWIG
20 FOR X=1536 TO 1578:READ
   A:POKE X,A:NEXT X
30 DATA 24,173,243,2,105,2,201,
   4,208,2,169,0,141,243,2,
   169,8,141,26,2,96
40 DATA 169,0,141,27,2,141,40,
   2,169,6,141,26,2,141,41,2,
   169,64,141,14,212,96
50 POKE 539,0:POKE 538,2:POKE
   552,0:POKE 553,6:POKE
   54286,64:POKE 12,21:POKE
   13,6:POKE 9,1
```

```
0100 ;MIGAJACY KURSOR - MAC/65
0110 ;
0120 TIMVEC2 = $0228
0130 TIMCNT2 = $021A
0140 CHARCNT = $02F3
0150 NMIEN = $D40E
0160      *= $0600
0170 ;
0180 START CLC
0190 LDA CHARCNT
0200 ADC #$02
0210 CMP #$04
0220 BNE JUMP
0230 LDA #$00
0240 JUMP STA CHARCNT
0250 LDA #$08
0260 STA TIMCNT2
0270 RTS
0280 LDA # <START
0290 STA TIMCNT2+1
0300 STA TIMVEC2
0310 LDA # >START
0320 STA TIMCNT2
0330 STA TIMVEC2+1
0340 LDA #$40
0350 STA NMIEN
0360 RTS
```

Przedstawiony program może być pomocny przy pracy z komputerem, powoduje bowiem migotanie kursora, który staje się przez to bardziej widoczny i łatwiejsze jest redagowanie pisanego tekstu. Oprócz tego po każdym RESET ustawiane są zgodnie z życzeniem użytkownika tak ważne przy długiej pracy z komputerem parametry jak: kolor tła i ramki, jasność tekstu, lewy i prawy margines, a także szybkość wprowadzania znaków z klawiatury. Teraz naciśnięcie (nawet przypadkowe) klawisza RESET nie powoduje irytującej zmiany ustawionych parametrów.

Po uruchomieniu programu mamy migający kursor, tło ciemnozielone, ramkę jasnoniebieską, lewy margines 2, prawy = 37, czas do pierwszego powtórzenia klawisza = 15, szybkość powtarzania = 2. Jeżeli ustawione wartości komuś nie odpowiadają, można je zmienić dwoma sposobami: w samym programie w liniach DATA (trzeba również obliczyć nową sumę kontrolną i wstawić ją do linii 16 zamiast 8208), albo w trybie bezpośrednim za pomocą instrukcji POKE. A oto lista możliwych zmian:

POKE 1542,41	*— kursor migający*	*linia 20, liczba 41*
POKE 1542,9	*— kursor normalny*	
POKE 1543,x	*— częstotliwość migania*	*linia 20, liczba 8*
POKE 1572,x	*— lewy margines*	*linia 24, liczby 2 i 37*
POKE 1576,x	*— prawy margines*	
POKE 1580,x	*— jasność tekstu*	
POKE 1585,x	*— kolor tla*	*linia 26, liczby 12,192 i 148*
POKE 1590,x	*— kolor ramki*	
POKE 1595,x	*— czas do powtórzenia*	*linia 28, liczby 15*
POKE 1600,x	*— szybkość powtarzania*	*i 2 (po 169)*

Marek Krupa

```
10 REM * ACC  SZCZECIN '87 *
12 FOR A=1536 TO 1615
14 READ B:C=C+B:POKE A,B:NEXT A
16 IF C<>8208 THEN ? "POMYLKA
W LINIACH DATA !":END
18 A=USR(1554):NEW
20 DATA 169,2,133,9,165,20,41,8,
240,2,169,2,141,243,2,76,98,228
22 DATA 104,169,96,133,0,169,255
,133,8,169,35,133,2,169,6,133,3
24 DATA
169,2,133,82,169,37,133,83
26 DATA 169,12,141,197,2,169,192
,141,198,2,169,148,141,200,2
28 DATA
169,15,141,217,2,169,2,141,218
30 DATA
2,160,0,162,6,169,7,32,92,228
32 DATA 108,12,0
```

```
0100 ;Migajacy kursor - MAC/65
0110 ;
0120 BOOT? = $09
0130 CHARCTL = $02F3
0140 EXITVBL = $E462
0150 LNFLG = $00
0160 WARMST = $08
0170 CASINIT = $02
0180 LMARGIN = $52
0190 RMARGIN = $53
0200 COL1 = $02C5
0210 COL2 = $02C6
0220 COLBAK = $02C8
0230 KEYDEL = $0209
0240 KEYREP = $02DA
0250 SETVBL = $E45C
0260 DOSINIT = $0C
0270 RTCLOCK = $14
0280 ;
0290 *=$0600 ;poczatek procedury
0300 ;
0310 ;procedura VBLK
0320 ;
0330  LDA #$02    ;ustaw
0340  STA BOOT?   ;BOOT?
0350  LDA RTCLOCK
0360  AND #$08    ;migotanie
0370  BEQ EXIT    ;kursora
0380  LDA #$02
0390 EXIT STA CHARCTL
0400  JMP EXITVBL ;koniec
0410 ;
0420 ;procedura RESET
0430 ;
0440  PLA
0450  LDA #$60    ;ustaw
0460  STA LNFLG   ;rozkaz RTS
0470  LDA #$FF    ;ustaw
0480  STA WARMST  ;goracy start
0490 ;
0500  LDA #$23
0510  STA CASINIT ;ustaw
0520  LDA #$06    ;CASINIT
0530  STA CASINIT+1
0540 ;
0550  LDA #$02    ;ustaw
0560  STA LMARGIN ;lewego
0570  LDA #$25    ;i prawego
0580  STA RMARGIN ;marginesu
0590 ;
0600  LDA #$0C    ;jasnosc
0610  STA COL1    ;tekstu
0620  LDA #$C0    ;kolor
0630  STA COL2    ;tla
0640  LDA #$94    ;kolor
0650  STA COLBAK  ;ramki
0660 ;
0670  LDA #$0F ;czas pierwszego
0680  STA KEYDEL  ;powtorzenia
0690  LDA #$02    ;szybkosc
0700  STA KEYREP  ;powtarzania
0710 ;
0720  LDY #$00    ;ustaw
0730  LDX #$06    ;VBLK
0740  LDA #$07    ;uzytkownika
0750  JSR SETVBL  ;po systemowym
0760 ;
0770  JMP (DOSINIT) ;koniec
```

ATAK!

Tym razem coś dla miłośników łamania joysticka. Wystarczy wpisać program według zamieszczoneoo listingu i można strzelać do woli.

Na okołoziemskiej orbicie pojawili się "Obcy". Wkrótce okazało się, że ich zamiary wcale nie są przyjazne, a na dodatek dysponują potężnym uzbrojeniem. Bardzo szybko rozprawili się z flotami kosmicznymi wielkich mocarstw. Pozostałeś tylko Ty. Kierując bojowym kosmolotem "Gwiezdny Piorun" musisz pokonać Obcych. Jesteś ostatnią nadzieją Ziemi, jeśli zawiedziesz, wszyscy ludzie zostaną niewolnikami, a Ty zginiesz w mękach.

Po wpisaniu "RUN" program wykonuje kilka procedur i sam startuje. Gdy nie powiedzie Ci się lub gdy uzyskasz premię, gra będzie kontynuowana po naciśnięciu klawisza "START" lub przycisku joysticka.

Opisu gry nie będzie. To dobre dla jajogłowych naukowców i znudzonych komputerem dzieci. Prawdziwy gracz nie zajmuje się takimi głupstwami. Ale jedna uwaga jest konieczna. Przed pierwszym uruchomieniem programu lepiej zapisać go na kasecie lub dyskietce. Są na świecie rzeczy, o których się filozofom nie śniło — a po co "wklepywać" wszystko jeszcze raz.

Wojciech Zientara

UWAGA: Nie należy wpisywać liter wydrukowanych przed numerem linii programu. NIE SĄ one częścią linii, lecz kodem kontrolnym "Edytora BASIC-a" (zob. "Bajtek" 1/88 lub "Bajtek-Atari").

```
SM 10 POKE 106,PEEK(106)-8:GOSUB 1000:GOS
   UB 2000:GOTO 3000
WA 100 FOR V=4 TO 19:POSITION OH,V-1:? #6
   ;"  ":POSITION OL,V-1:? #6;"  "
GS 110 POSITION Y,2:? #6;G$(1,16):POSITIO
   N Y,3:? #6;G$(18,34):POSITION Y,4:? #6
   ;G$(35):POKE 77,0
RO 120 POSITION 6,0:? #6;S;:POSITION H,V:
   ? #6;A$:POSITION L,V:? #6;B$
OS 130 POSITION P-2,20:? #6;" [illegible]  ":Y=Y+D
   :IF Y<1 OR Y>3 THEN D=-D
FU 140 IF STRIG(0)=0 THEN GOSUB 600:IF ((
   P=H OR P=H+1) AND A$<>"") OR ((P=L OR
   P=L+1) AND B$<>"") THEN GOSUB 300
HF 150 IF RND(0)>0.5 THEN GOSUB 500
UC 160 X=STICK(0):P=P-2*(X>8 AND X<12 AND
    P>2)+2*(X<8 AND P<17):OH=H:OL=L
WP 170 DI=(INT(RND(0)*2)+1)*ID:H=H+DI:IF
   H>18 OR H<0 THEN ID=-ID:H=18*(H>10):GO
   TO 170
WA 180 DI=(INT(RND(0)*2)+1)*JD:L=L+DI:IF
   L>18 OR L<0 THEN JD=-JD:L=18*(L>10):GO
   TO 180
GO 190 L=L+(L<2)-(L>18):IF A$="" AND B$="
   " THEN 210
GG 200 POSITION 0,22:? #6;W$(1,W):NEXT V:
   IF A$="[illegible]" OR B$="[illegible]" THEN GOSUB 400
LU 210 IF J<6 AND I<6 AND A$="" AND B$=""
    THEN 700
DO 220 IF ABS(P+1-L)>1 AND ABS(P+1-H)>1 T
   HEN 3020
KP 230 IF (ABS(P+1-L)<2 AND B$="") OR (AB
   S(P+1-H)<2 AND A$="") THEN 3020
YI 240 FOR N=1 TO 20:POSITION P,20:? #6;"
   [illegible]":POSITION 5,10:? #6;" WYNIK:";S:SOUN
   D 0,150,8,12:FOR M=1 TO 20:NEXT M
HD 250 POSITION P,20:? #6;"[illegible]":POSITION 5,
   10:? #6;" WYNIK:";S;" ":SOUND 0,100,8,
   12:FOR M=1 TO 20:NEXT M:NEXT N
LN 260 SOUND 0,0,0,0:IF A<S THEN A=S
TB 270 S=0:Q=3:GOSUB 900:GOTO 3000
FY 300 IF P=H+1 OR P=H THEN POSITION H,V:
   ? #6;"''":A$=""
RT 310 IF P=L+1 OR P=L THEN POSITION L,V:
   ? #6;"''":B$=""
BM 320 FOR N=300 TO 0 STEP -1:SOUND 0,80,
   8,N/20:NEXT N:S=S+1:RETURN
PF 400 IF A$="[illegible]" THEN G$(J,J+1)="[illegible]"
NR 410 IF B$="[illegible]" THEN G$(I,I+1)="[illegible]"
ZE 420 RETURN
BD 500 C=H:IF RND(0)<0.5 THEN C=L
NZ 510 IF C=H AND A$="" OR C=L AND B$=""
   THEN RETURN
XN 520 FOR N=V+1 TO 19:SOUND 0,10*N,4,10:
   POSITION C,N:? #6;"::":NEXT N
HL 530 FOR N=V+1 TO 19:SOUND 0,200,4,20-N
   :POSITION C,N:? #6;"  ":NEXT N
DQ 540 SOUND 0,0,0,0:IF C=P OR C+1=P THEN
    240
ZL 550 RETURN
OD 600 FOR N=19 TO 5 STEP -1:SOUND 0,2*N,
   10,10:POSITION P,N:? #6;"+":NEXT N
SJ 610 FOR N=19 TO 5 STEP -1:SOUND 0,5,10
   ,N*0.5:POSITION P,N:? #6;" ":NEXT N
ZG 620 RETURN
QZ 700 W=W+1:S=S+5:FOR N=1 TO 20:POSITION
    6,2:? #6;" premia ":FOR M=1 TO 20
FY 710 NEXT M:POSITION 6,2:? #6;" premia
   ":FOR M=1 TO 20:NEXT M:NEXT N
EJ 720 W=W-(W>20):GOSUB 900:GOTO 3010
CE 900 ON PEEK(53279)<>6 AND STRIG(0) GOT
   O 900:RETURN
OU 1000 GRAPHICS 0:SETCOLOR 2,0,0:POKE 82
   ,1:POKE 752,1:POSITION 1,5
FB 1010 ? "[illegible]":? "[illegible]"
DB 1020 ? "[illegible]":? "[illegible]"
ZU 1030 ? "[illegible]":? "[illegible]"
GM 1040 ? "[illegible]":POSITION 6,20:? "(c) 1987,
   Wojciech Zientara":POKE 82,2
AO 1050 RETURN
WY 2000 DIM W$(20),G$(49),A$(2),B$(2)
HL 2010 CHB=PEEK(106)+4:A=USR(ADR("hh[illegible]"),57344,256
   *CHB):RESTORE 10000
VF 2020 FOR I=0 TO 6:READ A:FOR J=0 TO 7:
   READ D:POKE J+8*A+256*CHB,D:NEXT J
QX 2030 NEXT I:A=0:S=0:RETURN
YA 3000 W$="&&&&&&&&&&&&&&&&&&&&":W=1
HK 3010 G$=" [illegible]     [illegible]     [illegible] ":D=1:Y=0:P=8
WN 3020 FOR I=48 TO 3 STEP -3:IF G$(I,I)=
   " " THEN NEXT I:I=3
NI 3030 H=I-19*(I>19)-19*(I>38)+Y:G$(I-1,
   I+1)="   ":IF I<4 THEN 3070
DJ 3040 ID=(H<10)-(H>9):FOR J=48 TO 3 STE
   P -3:IF G$(J,J)=" " THEN NEXT J:J=3
IE 3050 L=J-19*(J>19)-19*(J>38)+Y:G$(J-1,
   J+1)="   ":JD=(L<10)-(L>9)
BO 3060 OH=H:OL=L:B$="[illegible]"
BI 3070 A$="[illegible]":GRAPHICS 17:POKE 756,CHB:
   SETCOLOR 2,12,4:SETCOLOR 0,3,8:SETCOLO
   R 3,0,14:SETCOLOR 1,5,6
KB 3080 POSITION 0,0:? #6;"wynik:";S;"
   ":POSITION 10,0:? #6;"rekord:";A;" ";
LK 3090 POSITION 0,22:? #6;W$(1,W):GOTO 1
   00
UN 10000 DATA 3,2,2,71,125,7,31,59,49,32,
   64,64,226,190,224,248,220,140
YD 10010 DATA 4,49,59,31,7,125,71,2,2,5,1
   40,220,248,224,190,226,64,64
AC 10020 DATA 1,24,60,153,165,255,255,219
   ,129,7,145,82,0,3,192,0,74,145
RR 10030 DATA 6,128,192,224,192,128,128,1
   28,128
```

Polskie Znaki

Temat polskich znaków był już w KLANIE COMMODORE poruszany niejednokrotnie. Podawany był sposób ich definiowania dla Commodore 64, podawany był także dla C-128 w trybie 40 znaków. Poniżej przedstawiamy Czytelnikom dwa artykuły kończące niejako ten cykl — polskie znaki dla C-16/116+4 oraz dla C-128 w trybie 80 znaków.

C128

Mikrokomputery, którymi się posługujemy nie są niestety standardowo zaopatrzone w polski zestaw znaków. Stanowi to często poważne utrudnienie, gdy chcemy by nasz skądinąd świetny program komunikował się z nami w naszym rodzimym języku.

Na tamach BAJTKA pojawiały się już programy umożliwiające przedefilowanie znaków np. w trybie 40 kolumn w komputerze Commodore 128. Wszystkie te metody mają jednak jedną kolosalną wadę: zabierają dużo cennego miejsca w pamięci bo jak wiemy generator znaków, żeby móc go przedefiniować, należy najpierw przepisać z ROM na RAM. Komputer C-128 może pracować w dwóch trybach: 40 i 80 kolumn. W trybie 40 kolumnowym ekran jest kontrolowany przez układ VIC 8564. Jest to układ prawie identyczny (oprócz dwóch dodatkowych rejestrów) ze znanym z C-64 układem VIC 6564. Używając tego trybu pracy możemy korzystać zarówno ze zwykłego monitora jak i telewizora. Ponieważ komputer C-128 jest już komputerem półprofesjonalnym, ma on możliwość korzystania z drugiego trybu pracy (80 kolumn). Daje to użytkownikom możliwość wygodnego korzystania z całej gamy programów użytkowych jak edytory tekstów, bazy danych oraz z programów pracujących pod kontrolą systemu operacyjnego CP/M. Do tego stało się jednak konieczne zainstalowanie nowego układu VDC 8563 (Video Display Controller). W tym wypadku możliwe jest jednoczesne podłączenie do komputera monitora RGB oprócz jednego z już wyżej wymienionych odbiorników.

Przedstawiony poniżej program oprócz tego, że umożliwia korzystanie z polskich znaków w trybie 80 kolumn, NIE ZUŻYWA ANI JEDNEGO DODATKOWEGO BAJTU PAMIĘCI. Znaczy to tyle, że po wprowadzeniu i uruchomieniu tego programu będziemy dysponowali polskim zestawem znaków oraz taką samą pamięcią jak po normalnym włączeniu komputera. Dzieje się tak, ponieważ w odróżnieniu od VIC 8564, VDC 8563 posiada tylko dwie lokacje w pamięci poprzez które możemy się odwoływać do 37 rejestrów tego układu (adresy ŚD600 i ŚD601). Oznacza to, że w całej pamięci komputera są tylko dwie komórki poprzez które system operacyjny oraz my mamy wpływ na działalność tego układu oraz na dodatkowe 16K własnej RAM, zupełnie niezależnej od komputera. Właśnie w tym obszarze przechowywany jest zestaw znaków (zupełnie odmienny od tego jakim posługuje się komputer na ekranie 40 kolumnowym). Cecha ta pozwala przedefiniować znaki, a zarazem zaoszczędzić trochę miejsca. No tak, ale jak to zrobić mając bezpośredni dostęp tylko do dwóch rejestrów bez możliwości posłużenia się instrukcjami POKE i PEEK? Pierwsza część przedstawionego programu (do linii 190 stanowi właśnie rodzaj rozbudowanej instrukcji POKE, która umożliwia odwoływanie się do VDC 8563. Druga część programu zajmuje się natomiast definicją polskich liter. Po uruchomieniu programu polskie znaki uzyskujemy następująco:

CMD -a =ą
CMD-z =ź
CMD-c =ć
CMD-n =ń
CMD-l =ł
CMD-e =ę
CMD-s =ś
CMD-o =ź
CMD + =ż

CMD oznacza klawisz COMMODORE w lewym dolnym rogu klawiatury.

Do zdefiniowania znaków w trybie 80-kolumnowym VDC 8563 rezerwuje 2x64 bitów na każdy znak, chociaż istniejący zestaw jest zdefiniowany na matrycach 8x8 bitów. Taki tryb rozmieszczenia kolejnych znaków, oraz fakt, że VDC 8563 może jednocześnie korzystać z 512 znaków graficznych i literowych powoduje, że należy korzystać z następujących wzorów dla określenia pozycji znaku, który chcemy przedefiniować:

ASC * 16 + 8192 dla trybu "dużych liter" (uppercase). ASC * 16 + 12288 dla trybu "małych liter" (lowercase). W liniach 340^ł40 pierwsza wartość to kod ASCII przede- finiowywanego znaku. Pozostałe wartości w każdej linii to dane określające nowy znak.

Dominik Falkowski

```
36 115 REM **              87/12/28              **
78 120 :
B9 125 :
55 130 FAST:PRINTCHR$(14)
D0 135 FOR X=6144 TO 6182
BE 140 READ A:DF=DF+A:POKE X,A
EB 145 NEXT
42 150 IF DF<>3411 THENPRINTCHR$(7);:LIST:END
B3 155 DATA 072,138,072,152,072,169,002,141,040
B2 160 DATA 010,162,018,104,032,027,024,232,104
D1 165 DATA 032,027,024,162,031,104,076,027,024
7B 170 DATA 142,000,214,044,000,214,016,251,141
6F 175 DATA 001,214,096
5F 180 RESTORE 250
09 185 DO
FA 190 READ FG
0A 195 IF FG=-1 THEN EXIT
E5 200 RT=FG*16+12288
BD 205 FOR X=0 TO 7
28 210 AD=RT+X
55 215 HI=INT(AD/256):LO=AD-(HI*256)
9A 220 READ CH
DF 225 SYS 6144,CH,LO,HI
0B 230 NEXT X
FE 235 LOOP
2B 240 PRINT" f . J . 1 r i I 1 ' I "
2E 245 NEW:END
45 250 DATA 124,004,008,060,102,096,102,060,000
07 255 DATA 113,000,000,060,102,126,096,062,003
95 260 DATA 118,056,024,028,024,056,024,060,000
8C 265 DATA 106,004,008,124,102,102,102,102,000
E3 270 DATA 121,004,008,060,102,102,102,060,000
40 275 DATA 110,004,008,060,096,060,006,124,000
0B 280 DATA 109,004,008,126,012,024,048,126,000
5D 285 DATA 100,000,024,126,012,024,048,126,000
4D 290 DATA 105,096,096,112,096,224,096,126,000
2A 295 DATA 122,024,126,012,024,048,096,126,000
52 300 DATA 112,000,000,060,006,062,102,062,003
39 305 DATA -1
```

C-16/116/+4

Amator programowania stosunkowo prędko zacznie zastanawiać się, w jaki sposób wzbogacić czy uczynić* program bardziej zrozumiałym poprzez zastosowanie w nim polskich znaków diakrytycznych.

Generalnie rzecz biorąc we wszystkich komputerach. Commodore dokonuje się tego tak samo: oryginalne znaki są najpierw kopiowane z pamięci ROM generatora znaków do pamięci RAM, następnie użytkownik musi je odpowiednio zmodyfikować zgodnie ze swoimi potrzebami. Po tej operacji informuje się procesor graficzny skąd ma czerpać informacje o znakach i... gotowe.

Dla rodziny Commodore 16/116/+4 wszystkie funkcje spełnia poniżej przedstawiony program. Nowe znaki są w nim przypisane następującym klawiszom:

Znaki

ą — CMD + a   ć — CMD + c
ę — CMD + e   ł —CMD + l
ń — CMD + n   ó — CMD + o
ś — CMD + s   ź— CMD + z
ż — CMD + x   A —CMD + Q

Ć —CMD + Y   Ę —CMD + R
Ł — CMD + K   Ń —CMD + B
Ǫ — CMD + I   Ś —CMD + W
Ź —CMD + D   Ż —CMD + F

Znak "CMD" oznacza, że należy najpierw wcisnąć klawisz Commodore Logo Key (lewy dolny róg klawiatury) a następnie podaną literę.

Pierwsze i ostatnie pytanie zadawane nam przez program odnosi się do miejsca w pamięci w którym zostanie ulokowany nasz zestaw znaków. Posiadacze C-16/116 z pamięcią nierozszerzoną mogą wpisać tu wartości od 8 do 14; posiadacze PLUS/4 i C-16/116 z pamięcią rozszerzoną — wartości od 8 do 62. Im większa będzie podana przez nas ta wartość tym więcej będziemy mieli pamięci wolnej (nowy zestaw znaków zajmuje 1 KB pamięci). Klawisze funkcyjne f2 i f3 będą pełniły rolę przełącznika — wciśnięcie f2 powoduje wyłączenie zestawu polskich znaków podczas gdy f3 jego włączenie. Jeżeli podczas używania nowego zestawu znaków popełnimy błąd wpisując instrukcję czy rozkaz języka BASIC, to system operacyjny przełączy procesor graficzny tylko częściowo i ekran zablokuje się. Odblokować ekran można wciskając klawisz funkcyjny f2 lub f3.

Kazimierz Stróż

```
60 100 SCNCLR
E2 105 T$=CHR$(18)+"POL.LIT"+CHR$(146)
94 110 CHAR 1,16,3,T$
F5 115 CHAR 1,4,7,"PROGRAM GENERUJACY POLSKIE ZNAKI"
C4 120 CHAR 1,5,9,"DLA COMMODORE 16, 116 I PLUS/4"
2B 125 PRINT:PRINT
1B 135 INPUT "LOKALIZACJA : ";H%
9B 140 H%=4*H%
5A 145 POKE 55,0:POKE 56,H%
58 150 HX=PEEK(56):ZN=54272
5A 155 A$="POKE65299,"+STR$(H%)
F5 160 A$=A$+":POKE65298,192:POKE1351,128"+CHR$(13)
EF 165 KEY 3,A$
98 170 A$="POKE65299,208:POKE65298,196:POKE1351,0"
EB 175 KEY 2,A$+CHR$(13)
4B 180 POKE 1177,62
C5 185 FOR I=0 TO 1023
C8 190 POKE 256*H%+I,PEEK(ZN+I)
AF 195 NEXT
9C 200 POKE 1177,63: POKE 1351,128
50 205 POKE 65298,192: POKE 65299,H%
47 210 FOR J=1 TO 18 : READ B
E3 215 FOR I=0 TO 7 :READ A
6D 220 POKE 256*H%+8*B+I,A
CE 225 NEXT:NEXT
4F 230 :
5A 235 SCNCLR
EF 240 CHAR 1,6,10,"POLSKIE ZNAKI ZAINSTALOWANE    "
14 245 PRINT:PRINT:PRINT
E5 250 END
12 255 :
83 260 DATA 112,000,000,060,006,062,102,062,004
FD 265 DATA 113,000,000,060,102,126,096,060,008
E3 270 DATA 121,004,008,060,102,102,102,060,000
B8 275 DATA 124,004,008,060,102,096,102,060,000
4C 280 DATA 106,004,008,124,102,102,102,102,000
C9 285 DATA 109,004,008,126,012,024,048,126,000
31 290 DATA 125,008,000,126,012,024,048,126,000
46 295 DATA 118,056,024,028,024,056,024,028,000
90 300 DATA 110,004,008,060,096,060,006,124,000
1A 305 DATA 107,024,060,102,126,102,102,102,008
BF 310 DATA 114,126,096,096,120,096,096,126,004
BB 315 DATA 098,060,106,098,098,098,098,060,000
4F 320 DATA 126,060,106,096,096,096,102,060,000
69 325 DATA 127,106,110,118,126,110,102,098,000
73 330 DATA 108,126,006,012,126,048,096,126,000
63 335 DATA 123,126,038,012,024,048,096,126,000
F8 340 DATA 097,096,104,112,096,224,096,126,000
FD 345 DATA 115,060,106,096,060,006,102,060,000
```



PAPIEROWY IBM

Nie ma sensu chyba wspominać czym jest firma IBM na rynku komputerowym, gdyż nie ma chyba entuzjasty komputerów, który by o niej nie słyszał. Komputer jednak to nie tylko sprzęt, konieczna jest także dla niego szczegółowa dokumentacja bez której niemożliwe byłoby wykorzysta- nie wszystkich jego możliwości. W tej zaś dziedzinie odpowiednikiem IBM na rynku wydawniczym jest zachodnioniemiecka firma DATA BECKER.

DATA BECKER powinna być dobrze znana polskim czytelnikom, a zwłaszcza entuzjastom mikroinformatyki. Swoją działalność firma rozwija w trzech głównych kierunkach: literatury komputerowej o charakterze ogólnym, literatury o charakterze inżynierskim i elektronicznym oraz produkcji oprogramowania. Z chwilą rozpoczęcia się boomu komputerowego rynek stał się bardzo chłonny na wszelkie publikacje dotyczące mikrokomputerów. Sytuację tę wykorzystał znakomicie szef firmy, dr Achim Becker tworząc podwaliny pod potężne przedsiębiorstwo wydawnicze, które ma już na swym koncie znaczący dorobek. Publikacje ta- , kie jak zasady programowania w danym języku, czy programowanie w asemblerze znikały szybko z półek księgarskich, ź biegiem czasu jednak czytelnik zaczął poszukiwać publikacji i książek bardziej zaawansowanych omawiających mapy pamięci poszczególnych modeli mikrokoputerów, architektury systemów operacyjnych, zasadę działania z punktu widzenia współpracy poszczególnych układów scalonych itp. Elastyczne podejście do wymagań rynku spowodowało, że DATA BECKER zanotowała następny sukces. Powstało wtedy wiele książek o tej tematyce — np. ATARI ST INTERN, Das Grosse Floppybuch zur 1541 itp. Poszczególne modele komputerów i urządzeń peryferyjnych były "rozpruwane" przez specjalistów — elektroników pracujących dla firmy, a doświadczenia sprzedawane właśnie w formie książek z których sporo to prace zbiorowe.

Trzecim kierunkiem działalności jest oprogramowanie. Myślę, że większość posiadaczy Commodore czy Amstrada zna doskonale programy z końcówką "MAT", np. TEXTOMAT, DATAMAT, FAKTUMAT. DISKOMAT czy KONTOMAT. Większość z nich jest dostępna pod takimi samymi nazwami na wszystkich modelach komputerów od Commodore począwszy a na ATARI ST kończywszy. I znów trzeba tu wspomnieć o rozsądnym podejściu szefów firmy: zamiast na ilość DATA BECKER poszła na jakość. Przykładem może być tu np. TEXTOMAT 128, który (nie bez kozery!) jest uważany w Polsce za jeden z najlepszych edytorów tekstu.

Pieniądz robi pieniądz o czym zdają się doskonale wiedzieć szefowie DATA BECKER. Parę lat temu firma ta weszła na rynek amerykański podpisując umowę z firmą ABACUS SOFTWARE, która jest wyłącznym przedstawicielem DATA BECKER w USA. Współpraca ta przynosi wymierne korzyści nie tylko dla obu firm, ale także i dla czytelników w postaci coraz to nowych publikacji.

Poniżej chciałbym przedstawić czytelnikom jedną z książek DATA BECKER, którą redakcja otrzymała do recenzji. Książka ta. autorstwa Froitzheima i Kausmanna zatytułowana jest EFFEKTIV UND KREATIV i poświęcona jest komputerowi Commodore PLUS/4. Pierwsza część książki poświęcona jest oprogramowaniu zawartemu w pamięci ROM tego modelu i szczegółowo wyjaśnia w jaki sposób tworzy się bazy danych, drukuje faktury, miesza grafikę z tekstem czy przenosi dane z bazy do edytora tekstu. Omówienie to zawarte jest na 125 stronach z 244 składających się na całą publikację. W dalszych czterech rozdziałach czytelnik znajdzie 18 programów użytkowych napisanych w całości w BASIC co z kolei pozwala na ich dokładne przeanalizowanie. Programy te są poświęcone sortowaniu danych, obsłudze stacji dysków, tworzeniu "okienek", wykresów słupkowych i wykresów funkcji itp. Dla każdego coś miłego...

Do najciekawszych należy zaliczyć rozdział ostatni w którym przedstawiona została mapa pamięci Commodore PLUS/4 od adresu $0000 do $07FC (2044 dziesiętnie), oraz uwagi dotyczące wyboru i uzyskania kolorów. Taki dodatek (a zwłaszcza mapa pamięci) czyni książkę tę o tyle bardziej wartościową, że nawet jeśli komuś jest już BASIC dobrze znany, to będzie ona także pomocna przy programowaniu w języku wewnętrznym. Reasumując jest to publikacja starannie przemyślana pozwalająca czytelnikowi znacznie lepiej wykorzystać możliwości Commodore PLUS/4. Jeżeli dodatkowo weźmiemy pod uwagę, że panuje generalna posucha na literaturę i programy do tego modelu Commodore, książka ta okaże się na pewno bardzo przydatna.

FROITZHEIM KAUSMANN
EFFEKTIV UND KREATIV
Wydawca: DATA BECKER, 1985
244 strony, cena 49 DM,
ISBN 3-89011-073-8

Klaudiusz Dybowski.

PRZEDSTAWIAMY WARSAW BASIC

JAK ROZBUDOWAĆ INTERPRETER?

Użytkownik mikrokomputera klasy VIC-20 lub COMMODORE 64/128 bez specjalistycznego przygotowania informatycznego, jest prawdopodobnie przekonany, że napisanie interpretera to zadanie niewykonalne dla hobbystów.

Na szczęście, w naszym przypadku zabawa w pisanie interpretera nie wymaga tworzenia go od początku, gdyż wystarcza jego modyfikacja. Można ją wykonać metodą małych kroków. Daje to sporo satysfakcji, szczególnie w przypadku wspomnianych mikrokomputerów, bowiem CMB V2.0 jest ubogi, ale napisany bardzo elastycznie.

W rozpoczynanym dziś cyklu będziemy mówić i konkretnym rozwiązaniu dotyczącym konkretnego mikrokomputera. Chcemy przedstawić jak jest zrobiony i jak działa Warsaw BASIC. Czytelników, którzy będą próbowali porównywać możliwości tego interpretera BASICa z innym, prosimy o branie przy tym pod uwagę tylko tych ich wersji, które są zaimplementowane na podobnym sprzęcie.

Warsaw BASIC jest interpreterem w pełni profesjonalnym, instalowanym w mikrokomputerze COMMODORE 64, obejmującym wreszcie wszystkie czynności związane z kodowaniem, interpretacją kodu, obsługą przerwań, wyprowadzaniem danych i programu itp. W celu uproszczenia dalszych rozważań wydzielmy spośród wszystkich tych czynności, wykonywanych przez interpreter, dwie podstawowe: kodowanie i interpretację.

Kodowanie

Etap ten zaczyna się od podania komputerowi treści do przetworzenia i kończy się zapisaniem jej w odpowiednim miejscu w pamięci wewnętrznej komputera.

Jeżeli napiszemy na ekranie np.

(1) 100 a$= "Warsaw BASIC": print a$

i naciśniemy klawisz RETURN, to w odpowiednim miejscu w pamięci wewnętrznej (do tego jeszcze wrócimy!) w kolejnych "komórkach" pojawi się następujący ciąg kodów (zapisanych tutaj w systemie szesnastkowym):

(2) 64 00 41 24 b2 22 d8 41 52 53 41 58 20 c2 d c9 c3 22 3a 99 41 24 00

Numer linii 100 reprezentowany jest przez zawartość (w systemie szesnastkowym) nb=64 i wb=00 dwóch pierwszych "komórek", obliczoną według wzoru nb+wb*256=100.

Jak można zauważyć, następnym kolejnym znakiem w linii (1) odpowiada jeden kod w linii (2) za wyjątkiem ciągu znaków p r i n t. Zaznaczyła się tutaj działalność odpowiedniej procedury, która rozpoznała ten ciąg znaków i zakodowała go w postaci liczby 99 (w systemie szesnastkowym).

Interpretacja. Ten etap zaczyna się w chwili wykonywaniakomendy RUN i kończy się z chwilą zakończenia wykonania programu.

Po uruchomieniu programu uaktywnia się procedura, którą w dalszym ciągu będziemy nazywać pętlą interpretera, pobierającą i interpretującą zawartość kolejnych "komórek". Pętla interpretera rozpoznaje np. czy pobrany kod jest instrukcją, separatorem, końcem linii itp. i wywołuje, w zależności od wyniku tego rozpoznania, odpowiednią procedurę wykonującą zakodowane zadanie. Po wykonaniu tej procedury interpreter wraca do pętli.

W naszym przykładzie pętla interpretera pobiera kod litery a i sprawdza czy jest to kod jakiejś instrukcji. Ponieważ 41 nie jest kodem ani instrukcji ani separatora w interpreterze CMB V2.0 (lub CMB V7.0), to wywoływana jest procedura obsługująca instrukcję podstawienia (LET), która przejmie dalszą interpretację tego zapisu.

Po wykonaniu podstawienia pętla interpretera pobierze kod 3a (jest to kod separatora :). Gdyby w tym miejscu zabrakło dwukropka, to pętla interpretera odesłałaby do procedury obsługującej błędy (jak wiemy w tym miejscu program zakończyłby się komunikatem SYNTAX ERROR). Separator odsyła do pobrania zawartości następnej "komórki". W naszym przypadku będzie to kod 99 instrukcji PRINT.

Pętla interpretera kończy interpretację linii po napotkaniu kodu 00. W rozważanym przykładzie, po napotkaniu tego kodu i stwierdzeniu, że jest to ostatnia linia programu, interpreter zakończy wykonywanie programu.

Rozbudowa interpretera powinna zatem zawierać elementy, które umożliwią rozpoznanie słów kluczowych (instrukcji), ich kodowanie i rozszyfrowywanie na użytek pętli interpretera i wyprowadzania danych i treści programu z pamięci wewnętrznej.

Warsaw BASIC jest rozbudowanym interpreterem CBM V2.0 we wszystkich wymienionych kierunkach. Szczegółowa realizacja tak rozumianej rozbudowy nie jest oczywiście rzeczą prostą. Nie zważając na to, chcemy zapoznać Czytelników BAJTKA z wybranymi elementami Warsaw BASICa, niespotykanymi w innych interpreterach instalowanych w mikrokomputerach domowych i osobistych. Aby można było to uczynić w możliwie prosty sposób, omawiając w kolejnych odcinkach nie tylko zasady działania, ale i pokazując na przykładach działanie tego interpretera, jesteśmy zmuszeni do zawężenia modyfikacji interpretera do etapu interpretacji.

Mamy nadzieję, że zawężenie to pozwoli na zaprezentowanie w następnych odcinkach takiej modyfikacji interpretera CBM V2.0, która pokaże na czym polega dynamiczny podział pamięci wewnętrznej komputera, programowanie proceduralne i strukturalne, tworzenie biblioteki procedur itp. Aby to uczynić należy zacząć już w następnym odcinku od nauczenia pętli interpretera rozpoznawania nowych instrukcji.

K. Gajewski

CZWARTE WCIELENIE DINOZAURA

Niewiele jest na świecie komputerów ośmiobitowych, które przeżywają drugą młodość w postaci nowych wersji, natomiast na pewno istnieje tylko jeden, który pojawia się na rynku w swoim czwartym wcieleniu. Komputerem tym jest Commodore 64. W drugiej wersji zainstalowano go w innej, większej maszynie (C-128), w trzeciej — zmieniono mu wystrój zewnętrzny (C-64C), w czwartej zmieniono wnętrze.

Najnowsza wersja jest stosunkowo niedroga, a przy tym, dzięki wprowadzonym zmianom, bardziej niezawodna. Obudowa komputera pochodzi z C-64 natomiast klawiatura z wersji C-64C. Wnętrze komputera zostało całkowicie przeprojektowane i zmienione, a jedyną pozostałością są tu układy interface'ów CIA — 6526 i pamięć ROM oznaczona jako 901225.

System operacyjny oraz interpreter języka BASIC umieszczono w jednym układzie pamięci ROM oznaczonym jako 251913 co znalazło zastosowanie już wcześniej w Commodore 128. Rozkład wyprowadzeń tego układu jest taki sam jak układ wyprowadzeń popularnej pamięci EPROM 27128. Procesor 6510 został tu zastąpiony układem 8500 R4, którego lista rozkazów jest taka sama jak w poprzedniku.

Zupełną nowością jest zastosowanie nowego układu oznaczonego jako 251715 w którym zawarte jest całe 64 KB RAM. Jako ciekawostkę warto przytoczyć tu fakt, że układ ten może być zastąpiony układem o większej (np. 256 KB) pojemności co pozwala wykorzystać ją jako RAM-disk lub też jako normalną pamięć operacyjną dostępną dla procesora w postaci banków po 64 KB każdy.

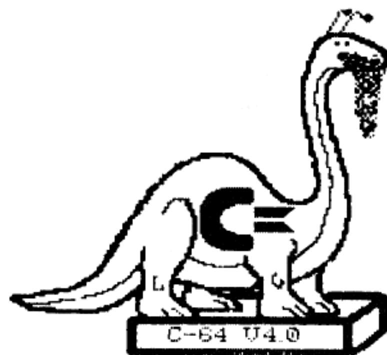
Wymieniony został także procesor graficzny i dźwiękowy — układy VIC 6569 oraz SID 6581. Nowy procesor graficzny (odpowiedzialny zarówno za tekst jak i grafikę) nosi oznaczenie 8565, układ dźwiękowy natomiast 8580. Firma Commodore tłumaczy wymianę tych układów tym, że do ich zasilania wymagane były dwa napięcia — 12V oraz 5V prądu stałego. Problemem było tu napięcie +12V wytwarzane z napięcia 9V prądu zmiennego pobieranego z regulatora napięcia umieszczonego w samym komputerze. Pomijając nawet fakt niezbyt stabilnego napięcia zasilającego delikatne układy, prowadziło to także do wytwarzania się dużych ilości ciepła. Problem ten rozwiązano w elegancki sposób — nawet układy są zasilane jedynie za pomocą 5V. Dodatkową zaletą takiego rozwiązania jest także obniżenie kosztów produkcji.

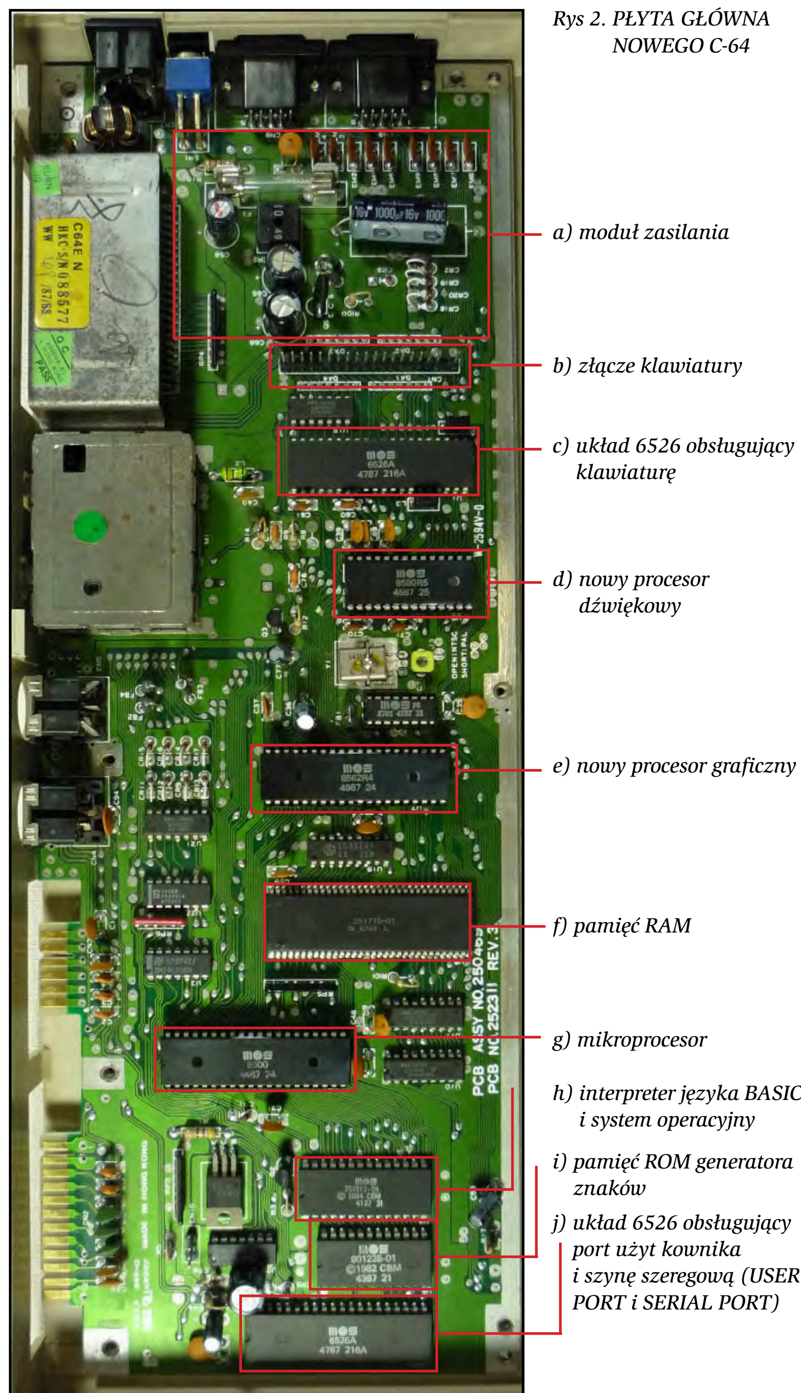
Kompatybilność nowej wersji C-64 ze starszymi wersjami jest stuprocentowa — dotyczy to również kompatybilności z rozszerzeniami przyłączonymi do portu rozszerzania (EXPANSION PORT) takimi jak FINAL CARTRIDGE, POWER CARTRIDGE, TURBO PROCESS, ACTION REPLAY itp.

Na koniec najważniejsza informacja, jego cena wynosi w RFN od 248—328 DM, (wrzesień/paźdź. 87).

Na podstawie "64'er" 09/87

Jan Jasiński
COMMODORE CLAN "KOMODA"





Rys 2. PŁYTA GŁÓWNA NOWEGO C-64

a) moduł zasilania
b) złącze klawiatury
c) układ 6526 obsługujący klawiaturę
d) nowy procesor dźwiękowy
e) nowy procesor graficzny
f) pamięć RAM
g) mikroprocesor
h) interpreter języka BASIC i system operacyjny
i) pamięć ROM generatora znaków
j) układ 6526 obsługujący port użyt kownika i szynę szeregową (USER PORT i SERIAL PORT)



ZEPSUŁEM PROGRAM NA TAŚMIE!

Sytuacja wydaje się beznadziejna. Zepsułem początek programu nagranego na taśmę. Nie mam listingu i w ten spo sób wiele godzin spędzonych przy pisaniu programu poszłona marne (nie licząc nabytego doświadczenia). Zepsuty jest tak nagłówek jak i sygnał pilotujący wraz z początkiem programu.

Otóż nie, jeżeli znasz dobrze swój komputer i sprawa jest warta tego, aby poświęcić jej więcej czasu. Na początku trochę wiadomości ogólnych o sposobie nagrywania na taśmie przez ZX Spectrum. Procedura ładująca i weryfikująca dowolny moduł z taśmy znajduje się w ROM-ie pod adresem #0556. Przed wywołaniem tej procedury należy umieścić w rejestrze IX adres początku wgrywanego bloku, w DE jego długość a w akumulatorze typ bloku (0 - nagłówek, 255-właściwy blok). Brak ustawienia znacznika C oznacza weryfikację. Nagrana na taśmie ilość bajtów jest zawsze o 2 bajty większa od tej długości, jaką deklarujemy w rejestrach DE. Na samym początku zapisywana jest zawartość rejestru A, co umożliwia sprawdzenie rodzaju bloku, oraz na samym końcu bajt parzystości budowany na bieżąco jako funkcja XOR z wszystkich nagrywanych bajtów. Pozwala to na końcowe sprawdzenie poprawności wczytanego bloku

W trakcie ładowania modułu do komputera jednym sprawdzianem poprawności ładowania jest odliczanie taktów zegara między kolejnymi zboczami napięcia przychodzącymi do portu #FE. Różnica czasu przekraczająca około 2400 taktów powoduje znany nam wszystkim komunikat "Tape loading error" już w trakcie wczytywania. Nie istnieje żadne inne sprawdzanie, czy aktualna para zboczy reprezentująca "0" lub "1" należy jeszcze do poprzedniego, czy już do następnego bajtu. Po odliczeniu 8 bitów komputer oczekuje następnej ósemki, itd. aż do osiągnięcia żądanej długości bloku.

Kłopot w tym, że opisana wcześniej procedura ładująca wymaga na początku sygnału pilotującego o większej odległości między zboczami oraz specjalnego impulsu synchronicznego, przed właściwym czytaniem bajtów. A przecież my właśnie to zniszczyliśmy! Może więc ominąć część tej procedury i przejść od razu do czytania poszczególnych bitów? Ale nie mamy wtedy żadnej pewności czy nasze czytanie rozpocznie się od początku bajtu! I jest to prawda. Z tym problemem postaramy się, przynajmniej częściowo, uporać później.

Na razie krótki program, napisany w assemblerze, umożliwiający wykorzystanie procedury ładującej z ominięciem opisanych wcześniej kłopotów.

```
ADRES   EQU adres       ; tu wstawić początek bloku
        ORG 65000
        LD IX, ADRES    ; ustawienie adresu początku
                          bloku w IX
        LD DE, #FFFF    ; maksymalna długość (i tak jej
                          nie znamy)
        LD C, #06       ; ustawienie koloru ramki
        XOR A           ; ustawienie znacznika Z, nie
                          będzie sprawdzana zgodność
                          pierwszego bajtu 5CF
                        ; ustawienie znacznika C,
                          ładowanie
        EX AF,AF'       ; przechowanie znaczników
                          w rejestrze F'
        DI              ; wyłączenie przerwań
                          maskowalnych
        LD HL,POWROT    ; odłożenie na stosie adresu
                          powrotnego
        PUSH HL           z procedury ładującej
        JP #05A3        ; skok do czytania pierwszego
                          bitu
POWROT  LD HL, #FFFF    ; obliczanie ilości wczytanych
                          bajtów
        SBC  HL,DE
        LD B,H          ; załadowanie ilości wczytanych
                          bajtów
        LD C,L            rejestrów BC
        EI              ; włączenie przerwań
                        maskowalnych
        RET
```

Przykładowy loader w ZX-Basic dla tej procedury przedstawia się następująco:

```
10 CLEAR 29999: LET adres =30000 20 FOR n = 0 TO
28: READ a: POKE 65500 + n, a: NEXT n: STOP
100 DATA 221, 33, adres-256*INT (adres/256), INT (adres/256)
110 DATA 17, 255, 255, 14, 6, 175, 55, 8, 243, 33, 240,
255, 229, 195, 163, 5
120 DATA 33, 255, 255, 237, 82, 68, 77, 251, 201
```

Po zainstalowaniu procedury ustawiamy magnetofon w miejscu gdzie jesteśmy pewni, że znajduje się już nasz odzyskiwany kawałek programu. Następnie włączamy magnetofon i wysyłamy w trybie bezpośrednim PRINT USR 65500. Istotna jest tutaj kolejność — najpierw włączamy magnetofon a później uruchamiamy procedurę. Wynika to z faktu, że procedura natychmiast przechodzi do czytania kolejnych bitów, co w przypadku niewłączonego jeszcze magnetofonu spowoduje powrót do ZX-BASIC. Istnieje także możliwość złego trafienia w przedziały czasowe nagrane na taśmie, co także zakończy się natychmiastowym powrotem do interpretera. Musimy wtedy przewinąć taśmę i ponownie uruchomić procedurę.

Po pozytywnym zakończeniu wgrywania na ekranie zostanie wprowadzona ilość wczytanych do komputera bajtów.Powróćmy teraz do problemu kolejności bitów w poszczególnych bajtach. Sytuacja jest najprostsza w przypadku programu napisanego w ZX-BASIC i nagranego bez wartości zmiennych. Wtedy znamy ostatni bajt (przed bajtem kontrolnym) jaki został nagrany na taśmę — jest nim liczba 13 (ENTER), która kończy każdy wiersz programu. Posłużmy się przykładem. Niech dwa ostatnie bajty wczytane z taśmy wynoszą 48 i 107. Rozkład bitów w tych bajtach oraz w liczbie 13 jest następujący (zaczernienie oznacza jedynkę)



Linią przerywaną zaznaczono lokalizację bajtu o wartości 13 w dwóch ostatnich wczytanych bajtach. Wynika z tego, że ostatni bajt powinien być przesunięty o trzy bity w prawo oraz trzy bity najbardziej znaczące winny być wzięte z poprzedniego bajtu. Funkcję przesunięcia o trzy bity możemy zrealizować dzieląc dany bajt przez 8. Aby przesunąć wszystkie bajty odzyskiwanego programu wystarczy napisać krótki program:

```
1000 LET końcówka = 0: FOR n^adres TO adreskonca
1010 LET bity przesuwane=INT (PEEK n/8): LET nowakoncowka = PEEK n-8*bity przesuwane
1020 POKE n, 32*końcówka+bityprzesuwane: LET końcówka=nowakońcówka: NEXT n
```

Musimy jeszcze ustalić prawidłowy początek naszego bloku. W tym celu przeglądamy początek wczytanego modułu zwracając uwagę, gdzie znajduje się liczba 13. Po niej powinna następować dwubajtowa liczba opisująca numer wiersza programu. Liczba 13 może jeszcze występować wewnątrz 5-cio bajtowej reprezentacji liczby w ZX-BASIC, po liczbie 14. Gdy zlokalizowaliśmy prawidłowy początek wiersza nagrywamy tak odzyskany moduł na taśmie jako moduł binarny instrukcją SAVE"nazwa"CODE adrespoczątku, adreskońca-adrespoczątku + 1. Pozostaje nam jeszcze tylko wgrać go ponownie pod programem COPY- COPY. Zmieniamy informacje zawarte w nagłówku:

1. wpisujemy do pierwszego bajtu 0,
2. wpisujemy do bajtu 14-tego wartość 32768,
3. wpisujemy do bajtu 16-tego długość modułu (wielkość podawana jako "dług" w trakcie wyświetlania nagłówków).

Po ponownym nagraniu na taśmie dysponujemy już odzyskanym fragmentem naszego "straconego" programu.

Niestety, w przypadku pozostałych modułów (tablice, binaria i programy w ZX-BASIC nagrana wraz ze zmiennymi) nie mamy żadnej informacji o ostatnim bajcie modułu. Pozostaje więc 7 możliwości przesuwania bitów w kolejnych bajtach, która z nich jest prawidłowa, możemy tylko próbować ocenić przyglądając zawartość pamięci po kolejnych przesunięciach kierując się własnym doświadczeniem i wiadomościami o tym, co powinien zawierać dany moduł. Oczywiście jest to wtedy bardzo pracochłonne i należy się zastanowić czy naprawdę warto odzyskiwać dany moduł o i tak zniszczonym bezpowrotnie początku.

Wojciech Indyk

KOMPUTER I TELEFON

Już od 1979 roku funkcjonuje we Frankfurcie nietypowy system telefonicznej informacji kolejowej. Zamiast ludzi zdecydowano się zatrudnić komputer. Jest on niezawodnym pracownikiem, który nigdy nie męczy się i zawsze jednakowo uprzejmie traktuje klientów. Podróżni zdążyli się już przyzwyczaić do jego monotonnego głosu i nadali mu zdrobniałe imię "Karolek".

Komputer udziela informacji 24 godziny na dobę. Prowadząc dialog pozdrawia on najpierw grzecznie pasażera, a następnie prosi o wykręcenie przy pomocy tarczy impulsowej telefonu kodu pocztowego miejscowości docelowej, daty podróży oraz przybliżonej godziny odjazdu. Karolek jest w stanie poinformować o połączeniach z 300 miejscami w całej Europie. Wszystkie niezbędne do tego dane są przechowywane w jego elektronicznej pamięci. System ten zyskał sobie duże uznanie u pasażerów.

Pomysł zastosowania tarczy telefonicznej jako urządzenia wspomagającego dialog człowieka z komputerem wykorzystano również w systemie przyjmowania zamówień. W tym przypadku należy wykręcić katalogowy numer towaru, który pragniemy zakupić.

(jj)

OD ŚRODKA cz. 2

Znacie już cały podział pamięci ZX Spectrum (mamy nadzieje, że zapamiętaliście coś z poprzedniego artykułu). A wiec do dzieła. Dziś poznany sposób zapisu programów na taśmie i zaczniemy sie włamywać do programów napisanych w BASIC-u (pamiętaj, że praktycznie każdy program ma chociażby procedurę ładująca napisana w BASIC-u. a jeśli jej nie posiada, to w zasadzie wcale nie jest zabezpieczony).

Na razie weźmy się jednak do sposobu zapisu programów na taśmie — przypomnij sobie co widać i słychać podczas wczytywania jakiegokolwiek programu.

Przez ok. 1 — 2 sekundy, na obrzeżu ekranu widoczne są szerokie, czerwono-niebieskie pasy oraz słychać ciągły dźwięk. Jest to tzw. pilot, który umożliwia komputerowi zsynchronizowanie się z sygnałem z taśmy, który ma odbierać. Potem przez krótką chwilę pojawiają się cienkie, migające, żółto-granatowe paski, oznaczające, że komputer wczytuje do pamięci informacje. Jest ich 17 bajtów — tzw. nagłówek. Pojawia się napis "Bytes:" "Program:", "Character array:" lub "Number array:", a potem po chwili przerwy zaczyna się drugi pilot (nieco krótszy), a po nim wczytuje się właściwy program. Zajmiemy się teraz nagłówkami, gdyż zawarte są w nich najważniejsze informacje o wczytywanych programach.

Nagłówek — jak już wspomnieliśmy — składa się z 17 bajtów. Będziemy je numerować od 0 do 16 (patrz rys.1). Zerowy bajt oznacza typ bloku. Jest on równy 0 — jeśli jest to program w BASIC-u, 3 — jeśli blok kodu maszynowego (nagrany przez **SAVE "CODE..."** albo **SAVE "..." SCREEN$**, które oznacza dokładnie to samo, co **SAVE "...",CODE 16384,6912**. Jeśli natomiast nagłówek ten poprzedza zbiór będący tablicą zmiennych BASIC-a (nagrany przez SAVE DATA...), to jest on równy 1 — dla tablic liczbowych lub 2 — dla tablic znakowych.

Następne 10 bajtów, to nazwa wczytywanego bloku, czyli tekst pojawiający się po załadowaniu nagłówka, za napisem "Program:", "Bytes:" czy innym.

Bajty 11. i 12. zawierają dwubajtową liczbę (pierwszy bajt mniej znaczący) oznaczającą długość bloku danych, którego ten nagłówek dotyczy.

W zależności od typu wczytywanego bloku, bajty od trzynastego do szesnastego są w różny sposób interpretowane.

Zacznijmy od nagłówków programów napisanych w BASIC-u. Bajty 13. i 14. oznaczają numer linii, od której ma się uruchomić program, jeśli był nagrany przez **SAVE "..." LINE** nr. Jeśli program był nagrany bez opcji LINE czyli po wczytaniu nie uruchomi się samoczynnie, to wartość tej liczby jest większa od 32767. Jednym ze sposobów unieszkodliwiania zabezpieczeń samostartujących programów w BASIC-u jest zmiana tych dwóch bajtów w nagłówku na wartość większą od 32767.

Bajty 15. i 16. zawierają liczbę oznaczającą długość samego programu w BASIC-u (bo SAVE "" lub SAVE "" LINE nagrywają program wraz ze wszystkimi zmiennymi, tzn. zawartość pamięci od bajtu wskazywanego przez zmienną systemową PROG, do bajtu określonego zmienną E_LINE). Gdy od całkowitej długości bloku (bajty 11. i 12.) odejmiemy tę liczbę, dowiemy się ile bajtów w tym bloku zajmują zmienne BASIC-a. To wszystko, jeśli chodzi o nagłówki programów w BASIC-u.

W nagłówkach kodu maszynowego ("Bytes:") bajty 15. i 16. nie są wykorzystywane, natomiast 13. i 14. tworzą dwubajtową liczbę określającą pod jaki adres należy wczytać następujący po nagłówku blok.

W nagłówkach tablic wykorzystywany jest spośród tych czterech, tylko bajt 14., który przedstawia nazwę wczytywanej tablicy. Jest on zapisany tak, jak nazwy wszystkich zmiennych BASIC-a (w obszarze od VARS do E_LINE), tzn. trzy najstarsze bity oznaczają typ zmiennej (tutaj będzie to tablica liczbowa lub znakowa) a pięć mniej znaczących bitów — nazwę tej zmiennej[*].

Na listingu 1. przedstawiony jest program tworzący procedurę umożliwiającą odczytywanie nagłówków programów zapisanych na taśmie. Po jego dokładnym wpisaniu (uważaj na właściwą ilość spacji w liniach z instrukcjami DATA!) uruchom go przez RUN i poczekaj chwilę. Na ekranie pojawi się informacja o adresach początku i końca procedury oraz o jej długości, która powinna wynosić 284 bajty. Jeśli tak nie jest, sprawdź, czy na pewno wpisałeś wszystkie linie programu. Jeśli wszystkie dane były poprawne, to na twojej taśmie znajdzie się procedura "czytacz", dzięki której będziesz mógł odczytać każdy nagłówek. Po nagraniu jej na taśmę możesz już skasować program z listingu 1.

Procedura "czytacz" jest relokowalna, tzn. będzie działała poprawnie niezależnie od tego, pod jakim adresem w pamięci się znajdzie. Możesz ją więc wczytać przez:

LOAD "czytacz" CODE adres

a następnie uruchamiać przez

RANDOMIZE USR adres

i to nawet wielokrotnie. Po uruchomieniu, procedura wczytuje pod adres 23296 (do bufora drukarki) pierwszy napotkany nagłówek. Jeżeli ze względu na podłączone urządzenia zewnętrzne nie odpowiada ci ten adres, to możesz go zmienić zastępując w linii 200 programu z listingu 1. liczbę 005B heksadecymalnym adresem, pod który chciałbyś wczytywać nagłówki (dwie pierwsze cyfry są młodszym bajtem tego adresu). Aby po takiej zmianie uniknąć sprawdzania sumy kontrolnej w tej linii, na końcu tekstu ujętego w cudzysłowy zamiast spacji i czterech cyfr sumy kontrolnej umieść literę "s" poprzedzoną czterema spacjami.

Po wczytaniu nagłówka procedura odczytuje informacje w nim zawarte i wraca do BASIC-a, lecz wczytanego nagłówka nie niszczy, więc jeśli chcesz, możesz go dodatkowo obejrzeć wykorzystując funkcję **PEEK**.

Samo odczytywanie nagłówków to jednak trochę mało, by móc włamywać się do bloków zapisanych na taśmie. Trzeba jeszcze wiedzieć, co z tymi blokami zrobić, aby umieścić je w pamięci, nie pozwalając im się przy tym uruchomić.

W przypadku bloków typu "Bytes" wystarczy zwykle załadować je pod wymuszony adres, powyżej RAM-TOP-u (tzn. powyżej komórki pamięci wskazywanej przez zmienną systemową RAMTOP), np. przez:

CLEAR 29999: LOAD ""CODE 30000

Metoda ta skutkuje, jeśli tylko wczytywany blok nie jest zbyt długi (może mieć maksymalnie ok. 40K). Dłuższe bloki mogą się po prostu nie zmieścić w pamięci — konieczne jest wówczas podzielenie ich na kilka krótszych części. Niedługo dowiemy się jak to robić.

Podobnie, w przypadku tablic, załadowanie ich nie nastręcza raczej kłopotów — wystarczy użyć normalnej w takich wypadkach instrukcji **LOAD "" DATA**...

Gorzej natomiast wygląda wczytywanie programów w BASIC-u. Są one zwykle nagrywane przez **SAVE "..."LINE**, a na samym początku linii, od której mają się uruchomić umieszczone są instrukcja zabezpieczające je przed zatrzymaniem. Najprostszym rozwiązaniem jest ładowanie programu nie przez LOAD"", lecz MERGE "", jednak metoda ta nie zawsze daje rezultaty. Z tej beznadziejnej wówczas sytuacji istnieją dwa zasadnicze wyjścia: podmienić nagłówek programu lub użyć przedstawionego niżej programu "load/merge".

Pierwsze z nich polega na zastąpieniu nagranego na taśmie nagłówka przez taki sam, ale nie powodują-

Bajt		
0	Typ	Typ bloku : 0 - Program, 1 - Number array, 2 - Character array, 3 - Bytes
1 2 3 4 5 6 7 8 9 10		NAZWA
11 12	Długość	długosc bloku danych - ile bajtów trzeba będzie zaraz wczytać z taśmy
13 14	Start	Bytes: adres ładowania bloku Program: numer linii od której program sie uruchomi Array: bajt 14-ty - nazwa zmiennej, 13-ty nieważny
15 16	Program	Program: długość samego BASIC-a, (bez zmiennych) Bytes, Array: nieważne

Rys. 1

cy uruchomienia programu. Można w tym celu użyć programu "COPY COPY" — wczytać nagłówek programu, wcisnąć BREAK, a następnie instrukcją LET zmienić jego parametr START na liczbę większą od 32767 (tzn. wykonać np. **LET 1 ="32768** — jeśli przerabiany nagłówek był wczytany jako pierwszy zbiór). Tak zmodyfikowany nagłówek nagrywamy gdzieś na taśmę. Usuwamy z pamięci program COPY COPY i wpisujemy LOAD "". Wczytujemy stworzony przed chwilą nagłówek, lecz zaraz po jego zakończeniu zatrzymujemy taśmę. Teraz do magnetofonu wkładamy kasetę z programem, do którego się włamujemy, ale ustawioną zaraz za nagłówkiem tego programu — tak, aby wczytać tylko treść programu, bez oryginalnego nagłówka.

Druga metoda jest o wiele wygodniejsza:

Wprowadzamy do pamięci (z klawiatury lub z taśmy) program "load/merge" zamieszczony na listingu 2. Po uruchomieniu zaczyna on czekać na pierwszy program w BASIC-u, który znajduje się na taśmie. Wczytuje go zupełnie tak samo, jak instrukcja LOAD "", lecz po załadowaniu nie pozwala na jego uruchomienie — drukuje jedynie komunikat "**0 OK**", z informacją od której linii wczytany program miał się uruchomić.

Na listingu 3. przedstawiona jest procedura, którą tworzy ten program poprzez POKE-i. Zasadniczą jej instrukcją jest CALL 1821. Począwszy od adresu 1541, w ROM-ie znajduje się procedura interpretująca instrukcje **SAVE, LOAD, VERIFY i MERGE. CALL 1821** wskakuje do środka tej procedury. Początkową jej część (odczytującą parametry tych instrukcji) pomijamy, zastępując to w liniach 40 — 90 naszej procedury: Najpierw przez RST 48 rezerwujemy 34 bajty w obszarze WORKSP**), adres wolnego miejsca przenosimy z rejestru DE do IX, w linii 80 zaznaczamy w zmiennej systemowej T__ADDR, że chodzi nam o **LOAD**, a nie np. **VERIFY**, a w pierwszym bajcie nagłówka do porównywania umieszczamy wartość 255, oznaczającą, że ma być wczytany pierwszy napotkany blok (tzn. program w BASIC-u). Reszty dokonuje procedura z ROM-u, działając identycznie jak **LOAD "".** Różnica w działaniu LOAD a "load/merge" polega na tym, że po całkowitym wczytaniu programu, zamiast wrócić przez **RET** do interpretera BASIC-a, przepisujemy zawartość zmiennej NEWPCC ("numer linii do której ma być skok") do PPC ("numer ostatnio wykonywanej linii") oraz nie dopuszczamy do uruchomienia programu — wykonujemy RST 8 z komunikatem "0 OK". Dzięki temu przepisaniu, w wydrukowanym komunikacie znajdzie się informacja o numerze linii, od której wczytany program miał się uruchomić.

Niecierpliwie czekając na kolejny odcinek z tego cyklu popróbuj odczytywać nagłówki bloków i wczytywać zabezpieczone programy w BASIC-u. Za miesiąc nimi właśnie dokładniej się zajmiemy.

Tomasz Surmacz
Robert Dudzik

```
Listing 1.
10 CLEAR 59999: LET pocz=60000
20 LET adr=pocz
30 RESTORE : READ a,b,c,d,e,f
40 DATA 10,11,12,13,14,15
50 LET nr=200: RESTORE nr
60 LET s=0: READ a$:  IF a$="." THEN GO TO
   130
70 FOR n=1 TO LEN a$-5 STEP 2
80 LET w=16*VAL a$(n)+VAL a$(n+1)
90 POKE adr,w: LET adr=adr+1: LET s=s+w
100 NEXT n
110 IF VAL a$(n TO )<>s THEN PRINT "blad
    w linii ";nr: STOP
120 LET nr=nr+10: GO TO 60
130 PRINT "wszystkie dane Ok."''"Poczatek
    :";pocz'"koniec : ";adr-1'"dlugosc :
    ";adr-pocz
140 SAVE "czytacz"CODE pocz,adr-pocz
150 REM
200 DATA "21920009E5DD21005BDDE51111 1246"
210 DATA "00AF37CD5605DDE130F23E02CD 1531"
220 DATA "011611C009DD7E00CD0A0CDDE5 1265"
230 DATA "D113010A00CD3C202A7B5CD1E5 1231"
240 DATA "D5ED537B5C010900CD3C20DD46 1346"
250 DATA "0CDD4E0BCD2B2DCDE32DE1DD7E 1664"
260 DATA "00DD460EDD4E0DC5A7202BEB01 1292"
270 DATA "1900CD3C20D5DD4610DD4E0FCD 1361"
280 DATA "2B2DCDE32DD1C178E6C0206EC5 1848"
290 DATA "011300CD3C20C1CD2B2DCDE32D 1280"
300 DATA "185EFE03203601700009EB0111  836"
310 DATA "0018E60D449C75676F9D9B2073 1281"
320 DATA "616D65676F2070726F6772616D 1313"
330 DATA "75200D4175746F737461727420 1161"
340 DATA "2D206C696E696120D2EE1B0181 1239"
350 DATA "0009EB010900CD3C20C13E1FA0 997"
360 DATA "F660D7DD7E003D28033E24D73E 1383"
370 DATA "28D73E29D7E1227B5C3E0DD7C9 1538"
380 DATA "04081C2020201C000010181030 268"
390 DATA "100C0008103840380478000D41 430"
400 DATA "64726573209C61646F77616E69 1357"
410 DATA "61200D5461626C69636120 862"
420 DATA "."
```

```
Listing 2.
1 REM       load/merge      TS & RD 1987
2 FOR n=60000 TO 60025: READ a: POKE n,a :
  NEXT n
3 RANDOMIZE USR 60000
4 DATA 1,34, 0,247,213,221,225,253,54,58,1,
  221,54,1,255,205,29,7,42,66,92,34,69,92,2
  07,255
```

* Sposób zapisu zmiennych BASIC-a w pamięci jest przedstawiony w dołączanym do Spectrum ang. lub niemieckim podręczniku na stronach 122 — 124, oraz w książce "Przewodnik po ZX-Spectrum" na str. 71.

** Pierwsze 17 bajtów z tych 34 zostaje zapełnione przez procedurę z ROM-u po odczytaniu parametrów wykonywanej instrukcji BASIC-a, drugie 17 to miejsce na wczytywany nagłówek. Obydwa te nagłówki sę następnie porównywane, zanim zostanie wczytany blok danych (sprawdzana jest zgodność nazwy i typu bloku).

```
Listing 3.
10 ;    load/merge
20 ;
30      ORG 60000
40      LD BC,34
50      RST 48
60      PUSH DE
70      POP IX
80      LD (IY+58),1
90      LD (IX+1),255
100     CALL 1821
110     LD HL,(23618)
120     LD(23621),HL
130     RST 8
140     DEFB 255
```



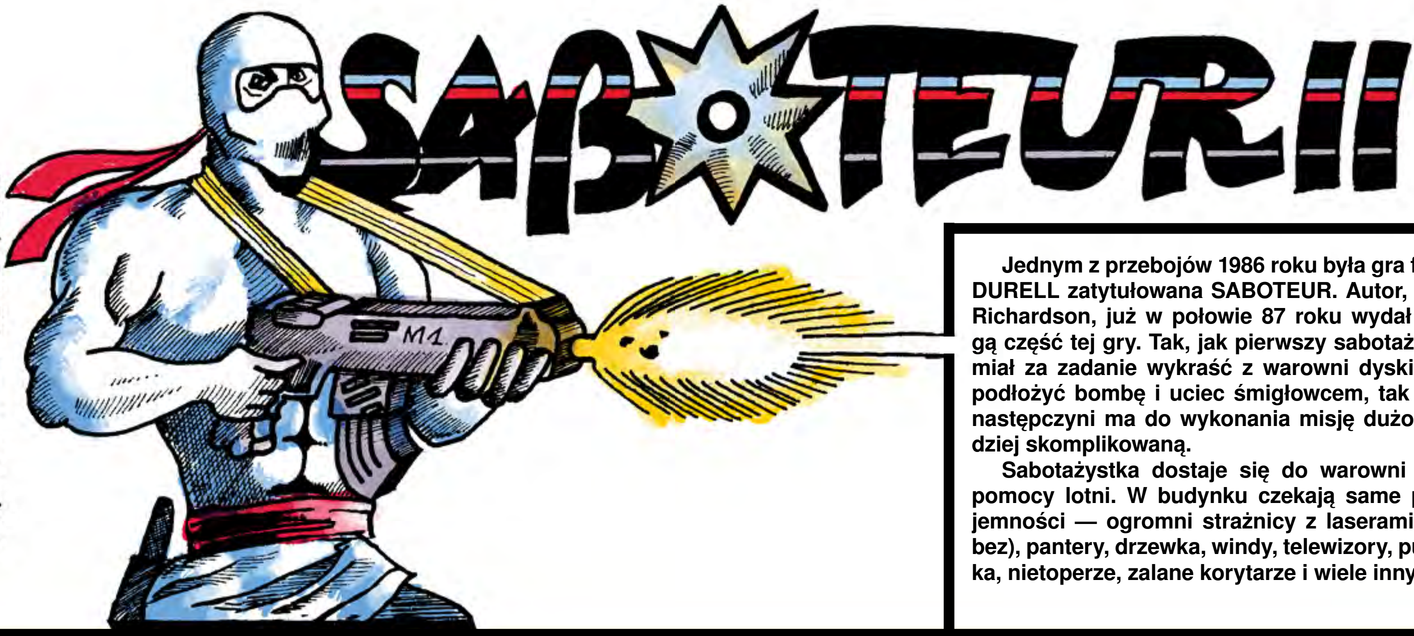
MÓJ MIKROKOMPUTER ZX SPECTRUM

Minęły już czasy, gdy poszukiwania jakiejkolwiek książki o mikrokomputerach z góry skazane były na niepowodzenie. Szczególnie dużej liczby publikacji doczekało się Spectrum. Kolejną pozycją skierowaną do sympatyków Sinclaira jest "Mój mikrokomputer ZX Spectrum" Rolanda Wacławka, wydaną przez Młodzieżową Agencję Wydawniczą.

Autor koncentruje się wyłącznie na BASIC-u. Opisane zagadnienia w sporej części pokrywają się z fabryczną instrukcją, ale można też znaleźć kilka użytecznych i oryginalnych ciekawostek. Książkę czyta się bardzo przyjemnie, nie nuży ona nadmiarem zbyt mądrych sformułowań. W przypadku literatury popularno-naukowej bardzo ważne jest, by była ona zrozumiała dla jak największej liczby czytelników. I to się autorowi udało. "Mój mikrokomputer ZX Spectrum" jest książką, od której można zacząć edukację. Zawiera informacje najważniejsze, zaprezentowane w formie gwarantującej, że nie zrazimy się dó komputera.

(j.j.)

R. Wacławek, "Mój mikrokomputer ZX Spectrum", Młodzieżowa Agencja Wydawnicza,Warszawa 1987, wydanie I, nakład 60 tys. egz. Cena 180 zł.



Jednym z przebojów 1986 roku była gra firmy DURELL zatytułowana SABOTEUR. Autor, Mike Richardson, już w połowie 87 roku wydał drugą część tej gry. Tak, jak pierwszy sabotażysta, miał za zadanie wykraść z warowni dyskietkę, podłożyć bombę i uciec śmigłowcem, tak jego następczyni ma do wykonania misję dużo bardziej skomplikowaną.

Sabotażystka dostaje się do warowni przy pomocy lotni. W budynku czekają same przyjemności — ogromni strażnicy z laserami (lub bez), pantery, drzewka, windy, telewizory, pudełka, nietoperze, zalane korytarze i wiele innych.

Na wszystko jedn
powiednio uderzon
pantera kończy żyw
drzewo można się w
łać windę, wyjąc z p
przed nietoperzem
stylem dowolnym.

Teraz coś o budy
kondygnacji połącz
mi w podłodze. Mar
dzy dwoma słupam
elektryczny, dwie w
motocykl, bardzo pr

Co właściwie ta z
leży to od wybrane
do 9. Do wybrania p
szy potrzebna jest
czeniu pierwszej mi
giej itd. Dla leni pod
misja 2 — JONIN, 3
— SAIMENJITSU.

To jeszcze nie k
misja polega na u

:MARTWE DRZEWO
:ŻYWE DRZEWO
:PANTERA
:LINA
:SKRZYNIE
:PRZEŁĄCZNIKI ŚWIATŁA ETC.
:PODŁOGA ETC.
:OKNO

START

U T S R Q P O N

K J I H G F E D

LIFT

MOTOCYKL

END

nak jest sposób. Strażnik od-
y pada na ziemię bez ducha,
wot po ciosie w szczękę, na
wdrapać, można też przywo-
udełka narzędzie, schylić się
i urządzić maraton pływacki

ynku. Zbudowany jest z kilku
onych drabinkami i... dziura-
my też linę rozciągniętą mię-
ni, rakietę na wyrzutni, płot
windy i podziemia. Stoi tam
rzydatny w ucieczce.
zrobić nasza bohaterka? Za-
ego poziomu trudności od 1
poziomu wyższego niż pierw-
znajomość hasła. Po ukoń-
isji otrzymamy hasło do dru-
daję kilka pierwszych haseł:
3 — KIMI, 4 — KUJI KIRI, 5

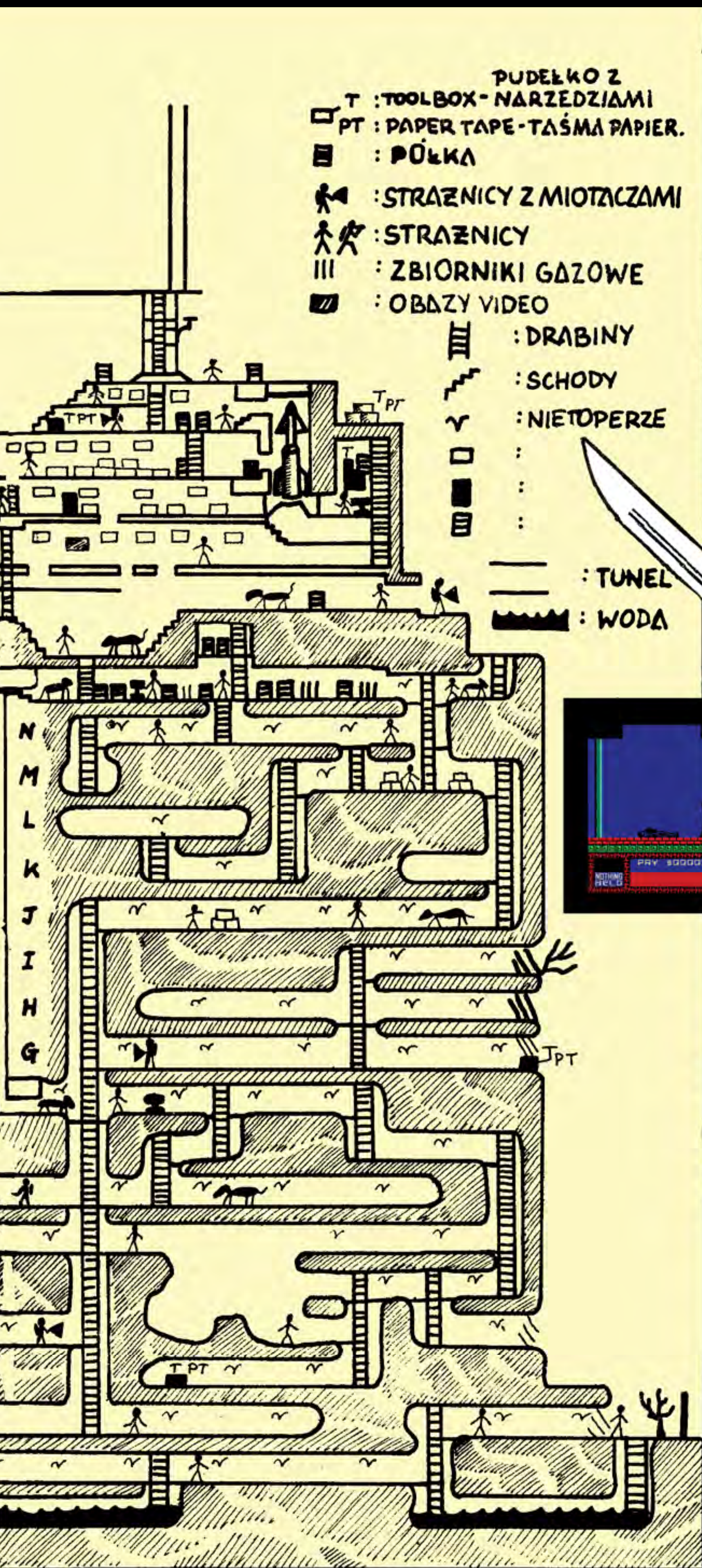
oniec urozmaiceń. Pierwsza
ucieczce motocyklem z bu-

dynku, druga — piechotą, w trzeciej trzeba już znaleźć kawałek taśmy perforowanej (ukryta w skrzynkach). Potem trzeba będzie wyłączać elektryczny płot i wykonywać wiele innych wymagających sprawnych rąk czynności. Ukończenie dziewiątej misji jest bardzo trudne, trzeba tam wykonać wszystko naraz (płot, 14 części taśmy itp.).

Tak, jak w pierwszej części, na ekranie pokazana jest zawartość kieszeni sabotażystki, ewentualne interesujące przedmioty w pobliżu, stan energii (właściwie siły witalnej) i zegar. Te dwa ostatnie wskaźniki można wyeliminować, ale o tym opowie POKErzysta na następnych stronach.

Autor: Mike Richardson
Firma: DURELL
Komputer: ZX Spectrum 48/+, Commodore 64/128, Amstrad/Schneider

(mp)



10 BAJTKOWA LISTA PRZEBOJÓW (2/88)

Drugie tegoroczne notowanie nie przyniosło rewelacji. Zniknął nawet BOULDER DASH, chyba już bezpowrotnie. Propozycji też jest niewiele — tylko 1911 na 103 tytuły.

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	POP EYE	↑		x	x	x
2	BARBARIAN	↑	x	x	x	x
3	NIGHT SHADE	↑		x		x
4	ANTIRIAD	↓		x	x	x
5	EQUINOX	!		x	x	x
6	MISS PACMAN	↓	x	x	x	x
7	CHEQUERED FLAG	!		x		x
8	GLADIATOR	↓	x	x		x
9	DAM BUSTERS	!		x	x	x
10	GREAT ESCAPE	↓		x	x	x

Nagrody – zestawy programów komputerowych otrzymują Joanna Molenda z Krakowa i Janusz Juszczak z Kalisza.



Kopertę nadesłał Adam Kiepuł z Łańcuta



KUNG-FU MASTER

Thomas wiele sobie obiecywał po pierwszej randce z Sylvią. Długo spacerowali po wąskich uliczkach Szanghaju. Kiedy zapadł zmrok, postanowili wracać do domu. Byli już blisko celu, gdy nagle z ciemnego zaułka wynurzył się podejrzany osobnik. W jego dłoni błysnęło ostrze noża. Reakcja Thomasa była natychmiastowa. Napastnik nie spodziewał się, że Thomas był adeptem szkoły wschodniej sztuki walki w klasztorze Shao-Lin. Dobrze wymierzone kopnięcie wytrąciło nóż z ręki bandyty, a silny cios w podbródek powalił go na ziemię. Nie był to jednak koniec walki. Zewsząd nadbiegały ciemne typki, by pomścić swego kolegę. Thomas zajęty walką nie zauważył, że zamaskowane postacie związały i uprowadziły Sylvię. W oka mgnieniu teren opustoszał. Thomas rozejrzał się szukając dziewczyny. Nie było ani jej ani napastników. Po całonocnym poszukiwaniu zmęczony dowlókł się do domu. Rozumiał już, co było celem napaści. Nie widział szans odzyskania ukochanej. Pogrążony w depresji myślał tylko o niej.

Pewnego dnia otrzymał list od pana X, z którego dowiedział się, że Sylvia porwana została przez starego i potężnego Ninję, i że jest więziona w ogromnym pałacu Ninji na najwyższym piętrze. Thomas natychmiast ruszył na pomoc ukochanej.

Pałac Ninji ukryty był wśród drzew. Thomas ostrożnie zbliżał się do wejścia. W pobliżu nie było żywej duszy. Mistrz ostrożnie wszedł do środka. Ledwo przestąpił próg, rozległ się trzask i brama zatrzasnęła się. W tej chwili uchyliły się drzwi w końcu sali. Thomas podbiegł do nich i zagłębił się w czeluść ciemnego korytarza. Powoli szedł naprzód omijając ukryte klapy-pułapki. Nagle zauważył sunącego prosto na niego mężczyznę. Nie miał wątpliwości co do jego zamiarów, uprzedził więc jego atak. Jak spod ziemi wyrósł przed nim następny napastnik lecz tego Thomas także powalił.

Podobnych niespodzianek spotkał jeszcze wiele. Po ziemi pełzały węże, znienacka wybuchały kolorowe ognie, dym gryzł w oczy a z ukrytych w ścianach drzwi wylatywały roje wirujących noży. Nic nie zdołało powstrzymać Thomasa przed nieustannym parciem naprzód. Pokonał w ten sposób cztery piętra i wreszcie znalazł się przed komnatą potężnego Ninji. Bez wahania wtargnął do środka. Tu czekała go najcięższa próba, z której też wyszedł zwycięzko. Mnisi z klasztoru Shao-Lin mogli być z niego dumni. A z Ciebie?

Firma: Ocean Software
Komputer: Commodore 64/128, Spectrum 48/+1+2, Amstrad/Schneider.

(m.p.)

KRÓL I KRÓLOWA GIER



JOASIA MOLENDA lat 12, zamieszkała w Krakowie, ul. Siewna 21a m 61, jest uczennicą Szkoły Podstawowej Nr 109. Jej hobby to pływanie i stąd chyba ulubioną grą Joasi jest SUBMISSION na ATARI 800 XL.



JANUSZ JUSZCZAK lat 14, zamieszkały w Kaliszu, ul. Grunwaldzka 21, chodzi do VIII klasy Szkoły Podstawowej. W domu ma Amstrada i ulubioną grę — SKOOL DAZE. W wolnych chwilach oprócz grania i programowania zajmuje się elektroniką.



BARBARIAN

Był słoneczny zimowy dzień. Mieszkańcy wioski byli zajęci swoimi sprawami. Nie przeczuwali zbliżającego się niebezpieczeństwa. Nagle z lasu wypadli nieznani wojownicy i rzucili się na mieszkańców. Część mieszkańców stawiła opór, krwawo stłumiony.

Rycerze spod znaku Węża zabili wszystkich mężczyzn i część kobiet i dzieci. Wśród pojmanych znajdował się mały Conan. Jego i paru innych chłopców wzięto do napędzania olbrzymiego koła. Ciężka praca już po roku wykończyła połowę ludzi. Po dziesięciu latach olbrzymie koło pchał tylko jeden człowiek — Conan.

Po pewnym czasie uwolniono go i skierowano do walk gladiatorskich, ku uciesze gawiedzi. Tu Conan nauczył się walczyć i już po kilku latach stał się niezwyciężonym wojownikiem. Darowano mu dzięki temu wolność. Wreszcie spełniło się marzenie Conana — być wolnym. W podświadomości czuł, że musi zemścić się na rycerzach spod znaku Węża.

I właśnie w tym momencie zaczyna się twoja rola. Musisz zabić wszystkich pilnujących cię dwudziestu wojowników Węża (Barbarian Game 1), w drugiej części po pokonaniu gwardii czarownika zmierzysz się z nim samym, by uwolnić więzioną przez niego dziewczynę.

Gra w wersji na ZX Spectrum ma możliwości wybrania 1 lub 2 graczy oraz opcji Kempston, Sinclair lub Keyboard. W lewym górnym rogu znajduje się twoja energia zaś w prawym — energia przeciwnika. Wygra ten kto dobierze drugiemu całą energię lub po prostu obetnie drugiemu głowę.

Barbarian czyli po polski barbarzyńca jest komputerową wersją filmu pt: "Conan Barbarzyńca".

Firma: Palące Software
Komputer: ZX Spectrum 48/+, Commodore 64//128, ATARI 800XL/520ST, Amstrad/Schneider.

(wso)

POKErzysta

Ostatnio w tej rubryce pojawiły się ułatwienia do gier zaliczanych do "nowości" i kopiowanych specjalnym interface. Poznać je można po tym, że podczas ładowania na ekranie znajduje się jeden z napisów: "M1 LOADING", "Jack OLantern presents" lub "BILL GILBERT PRESENTS". Tak oznaczona gra składa się z czterech zbiorów programu pilotującego, zbioru głównego, obrazka i zbioru dekodująco--startującego. Loader napisany jest w języku maszynowym i nie jest łatwo wpisać ułatwienia. Dla zaciętych graczy nie mających wiele wspólnego z programowaniem podam sposób "zapoke'owania" takiej gry na przykładzie PAPER BOY:

1. Załadować komenda MERGE "" zbiór oznaczony PAPER BOY i zatrzymać taśmę.
2. Napisać POKE 23838, 201: RUN (w przypadku BILL GILBERT — 23985,201).
3. Włączyć magnetofon.
4. Po załadowaniu całego programu pojawi się raport OK. Nie czyszcząc ekranu (sic!) wpisać z Bajtka POKE 48023,201.
5. Uruchomić grę komendą RANDOMIZE USR 24833. W ten sposóo poprawić można wszystkie gry w/w formatu.

A oto ułatwienia do innych, starszych gier:

ANDROID II	POKE 52262,0: POKE 53894,0
PYJAMARAMA	POKE 48670,0
EVERYONE'S A WALLY	POKE 28215,0
THE FALL GUY	POKE 27235,0
STOP THE EXPRESS	POKE 34464,183
ALIEN 8	POKE 45121 ,x — koniec gry
DUN DARACH	POKE 34999,255
TIR NA NOG	POKE 34202,255
MONTY MOLE	POKE 38004,0
CYCLONE	POKE 38291,50: POKE 63892,160: POKE 63893,146
DIGGER DAN	POKE 25559,0: POKE 26363,0
SAM SPADE	POKE 25215,0: POKE 26381,0
SPECTRAL PANIC	POKE 28522,0
STRANGE LOOP	POKE 63159,0: POKE 63161,0
COSMIC CRUISER	POKE 25373,0
KOKOTONI WILF	10 CLEAR 24100: LOAD ""CODE: LOAD "" CODE: RANDOMIZE USR 65100: LOAD ""CODE 30 POKE 28929,9: POKE 28934,8: POKE 28939,8: RANDOMIZE USR 41712
BLADE ALLEY	POKE 58201,0
FRED	POKE 31171,0
JETPAC POKE	25020,255: POKE 26075,0
BATTLERACE	POKE 44641,0
AQUARIUS	POKE 31055,0
ARCADIA	POKE 25776,0
NIGHT SHADE	POKE 58056,0: POKE 5744,0,0: POKE 53442,0: POKE 53443,12: POKE 51105,0
SABOTEUR II	POKE 37122,0: POKE 61340,201

Gracz

S.O.S.

Jak ukończyć grę BERK 3? Jak przejść komnatę z liną? W zamian za pomoc służę opisami gier HARBOUR ATTACK i FIRE ANT.
Henryk Tur, ul. Kazimierza W. 1/2, 37-700 Przemyśl

Mam ATARI 130 XE i nie umiem grać w grę UNIVERSAL HERO. Pomocy!
Paweł Kucharuk, ul. Dzierżona 35,44-100 Gliwice

Kto z czytelników przyśle mi dokładne plany i opisy do gier PYJAMARAMA i CHIMERA? Będę bardzo wdzięczny.
Ryszard Tabaka, ul. Budowlanych 19/10, 32-600 Oświęcim.

Posiadam ATARI 65 XE i następujące gry: KARATEKA, GREEN BERET, UNIWERSAL HERO, CAPTURE THE FLAG i ROAD RACE. Jednak potrzebne mi są opisy do nich. Proszę o pomoc.
Bartosz Jankowski, ul. Kolberga 19,41-710 Ruda Śląska 10 Wirek

Uprzejmie proszę czytelników Bajtka o przysłanie mi opisów gier: EUREKA 1, 2, 3, 4, 5. W zamian prześlę opisy ciekawych gier.
Jarosław Klima, ul. Sikorek 28A, 40-537 Katowice

Nie znam Poke'ów do gry EVERYONE'S A WALLY na Spectrum i proszę o pomoc.
Tomek Maćkowiak, ul. Klaudyny 14 m 64,01-684 Warszawa

Jak uzyskać nieśmiertelność w grach: MONTEZUMA'S REVENGE, SNOKIE, KARATEKA, SKY BLAZER, ROAD RACE, STAR FIGHTER, KANGAROO.
Marcin Prokop, ul. Szaserów 102/7, 04-335 Warszawa

Ludzie! Pomocy! Już od miesiąca w grze CAULDRON męczę wspaniałą czarownicę Orfidynę, która dzięki mnie traci wszystkie swoje życia. Proszę o przysłanie do tej gry nieśmiertelności. Czarownica będzie Wam wdzięczna. Komputer C64.
Paweł Koncewicz, ul. Rewolucjonistów 8/16,42-500 Będzin

Nie wiem, jak zakończyć grę AGENT USA i co robić, w tunelu w grze DIMENSION X na komputer ATARI.
Sławomir Micherda, os. 700-lecia 39/5, 34-300 Żywiec

Szukam opisu do gier AGENT USA, WARGAMES II, NIGHT CITY na ATARI 800 XL. W zamian oferuję opis do gry GHOSTBUSTERS.
Dominik Homa, ul. Cegielniana 24/16, 30-404 Kraków

Liczę na Waszą pomoc! Proszę o informacje na temat gier na ATARI: ACTION BIKER, CHIMERA, TIME SLIP, MONTEZUMA'S REVENGE.
Adam Czeczuk, ul. Grunwaldzka 15,17-100 Bielsk Podlaski

Mam problem z grami COLUMBIA i CHIMERA. W zamian proponuję opisy do gier; ONE ON ONE, TRANZ AM, FIGHTER PILOT, JET SET WILLY, KNIGHT LORE.
Bartosz Makswiej, ul. Lipowa 21a m 5,02-100 Węgrowice

Pomocy! Nie wiem, jak bawić się grami: MONKEY MAGIC, DOMAIN OF THE UNDEAD, MINER 2049 w wersji na ATARI 65 XE.
Tomek Szulc, ul. Obornicka 8/19, 91-039 Łódź Proszę o pomoc w grze WILD WEST na C64.
Adrian Nowak, ul. Sobieskiego 488, 42-580 Będzin 7

Bardzo proszę o przysłanie mi krótkich opisów gier: BEAR GEORGE, UNIWERSAL HERO, CAVELON. Mam Timex 2048.
Magdalena Chmielak, ul. Rodzinna 8/6, 57-300 Kłodzko

Pomóżcie mi rozpocząć grę B.C.BILL na ZX Spectrum.
Tomek Ufnarski, ul. M. Skłodowskiej 2,05-420 Józefów

Mieszkamy w Libii i nie mamy dostępu do żadnych opisów gier. Szukamy opisów do gier: ALIEN 8, SIR FRED, N.O.M.A.D., ACTION BIKER i ASTERIX. Bardzo chcielibyśmy nawiązać korespondencję z posiadaczami ZX Spectrum.
Łukasz Krzyżanowski, Krzysztof Zaremba, Polish Medical Staff, Po Box 77, Libya Zliten

Chciałbym posiadać grę ROLAND ON THE ROPES, lecz nigdzie nie mogę jej dostać. Będę bardzo wdzięczny za pomoc.
Radosław Kozielski, Pl. Wolności 7, 89-115 Mrocza

Bardzo proszę o przysłanie mi dokładnego planu gry MONTEZUMA'S REVENGE na mikrokomputer ATARI 800XL.
Mateusz Kasprzak, ul. 1-go Maja 67/17, 32-100 Proszowice

Potrzebny mi jest opis do gry UMBA oraz nieśmiertelności do gier: PYJAMARAMA i KNIGHT LORE.
Sebastian Miernik, ul. Próchnika 135/8, 53-407 Wrocław

Zwracam się do czytelników Bajtka o przysłanie mi krótkiego programu muzycznego na mikrokomputer Spectravideo 738 MSX.
Tomasz Zalewski, ul. Śniadeckich 25c/7,75-900 Koszalin

Posiadam "wnętrze" Spectrum 48 o oznaczeniu ISSUE 4B. Pilnie poszukuję obudowy z klawiaturą, najlepiej od Spectrum+. Ostatnia nadzieja w Czytelnikach Bajtka. Bardzo proszę o pomoc.
Tomasz Szymaniuk, ul. 1 Maja 19, 32-540 Trzebinia

Koniecznie potrzebuję opisy i sposoby uruchomienia gier: HOBBIT i DUN DARACH.
Maciej Pałka, ul. Walecznych 32 m 7, Wrocław

Szukam opisu gier: TECHNICIAN TED, SIR FRED, PYJAMARAMA, SCUBA DIVE, ELITE, BLUE THUNDER, BOOTY, MONTY MOLE.
Lech Kitlasz, ul. Czysta 22/1,16-010 Wasilków

Poszukuję programów Mega-Basic, Fifth, Blast, Toolkit, Mons 3, Gens 3. W zamian oddam Beta-Basic 3.0 i kompilator FP V I.7 oraz wiele innych.
Marcin Orłowski, ul. Radomska 38, 71-002 Szczecin

Proszę o pomoc w grach: SILENT SERVICE, KENNEDY APPROACH i MONTEZUMA'S REVENGE.
Michał Rosiak, ul. Jedlińska 5 m 1,02-929 Warszawa

Szukam nieśmiertelności do gier: RAMBO, COMMANDO, FIRELORD, FLYER FOX na Spectrum, a także wykazu programów symulacyjnych na ten komputer.
Bernard Adamczyk, ul. Jodkowa 1/6, 71-114 Szczecin

Poszukuję klawiszy sterowania do gier: RED MOON, FRANK'N'STEIN, HUNTER KILLER, DELTA WING, SPITFIRE 40, SOASH, YIE-AR KUNGFU, ARNHEM.
Wojciech Kacperski, ul. Chałubińskiego 19/61c, Gdynia-Chełm

Proszę o pomoc w grach na Commodore 64: W.A.R., SPY Vs SPY III, THE LAST NINJA (co zrobić ze smokiem ziejącym ogniem?).
Marek Ptaszyński, ul. Lumumby 101/105, 80-371 Gdańsk-Oliwa

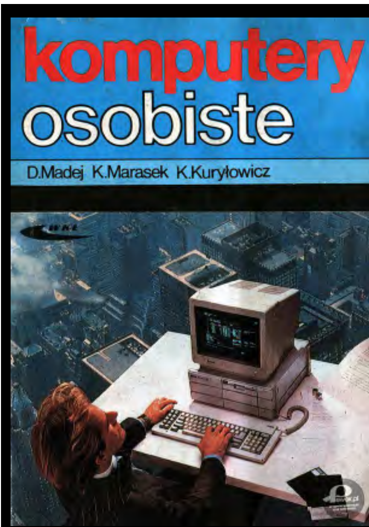
Pilnie poszukuję gier na ATARI: TIME-PILOT, GUN SMOKE, YIE-AR KUNGFU, KUNG-FU MASTER, PHOEENIX, COMMANDO.
Paweł Skrzypiński, ul. Dzierżyńskiego 6 m 17, 96-100 Skierniewice

Poszukuję dokładnego opisu programów gier na Atari 800 XL: MONTY MOLE, GHOSTS'N GOBLINS, PITFALL II.
Tomasz Bartosiewicz, ul. Fromborska 4a, 85-724 Bydgoszcz

Nie wiem, jak grać w COMBAT LYNX na Amstrada 6128. Potrzebne mi są też klawisze sterujące (używam klawiatury).
Jacek Fągowski, ul. Harcerska 16 m 36, 05-820 Piastów

Potrzebuję dokładny opis gry ZORRO na komputer ATARI (nie umiem wyjść z grobowca).
Maciej Mikulski, ul. Świerkowa 2, 44-100 Gliwice

Jestem użytkownikiem Spectrum+, proszę o pomoc w grach: BRIAN BLOODAXE, MONTY'S RUN, HEARTLAND, CHILLER, PYJAMARAMA. W zamian oferuję kserokopie opisów do NODES OF YESOD, ROBOT MESSIAH, COBRA STALLONE, TRAP DOOR.
Krzysztof Dęga, ul. Bydgoska 61c m 3, 64-920 Piła 4



KOMPUTERY OSOBISTE

Z Wydawnictwami Komunikacji i Łączności związali się na dłużej trzej autorzy "Przewodnika po ZX Spectrum" Dariusz Madej, Krzysztof Marasek i Krzysztof Kuryłowicz. Kolejne ich dzieło — "Komputery osobiste" to skarbnica wiedzy o mikrokomputerach. Książka ta może być znakomitym prezentem również dla kogoś, kto nie ma zamiaru spędzać połowy dnia przy klawiaturze. Znajdziemy w niej ogromną ilość wiadomości i odpowiedź na wiele pytań, które intrygują nie tylko informatyków. Przyda się też rodzicom, których dziecko w każdej chwili może poprosić o wyjaśnienie, co to jest mikrokomputer. Zawiera obszerne komentarze poświęcone sprzętowi, językom programowania, programom użytkowym. Ułatwia zrozumienie, że komputer jest bardzo wyrafinowanym urządzeniem, którego zastosowanie nie kończy się tylko na grach.

Wydawnictwu i autorom należą się gratulacje, że zdążyli umieścić w książce informacje niemal z ostatniej chwili, o twardych dyskach, dyskach optycznych, drukarkach laserowych, nowoczesnych pakietach oprogramowania czy najnowszym sprzęcie.

Dużą pomocą dla czytelnika są bardzo ładne i czytelne ilustracje, które często więcej mówią niż kilkustronico- wy wywód. Znakomitym tego przykładem jest np. rozdział poświęcony sieciom lokalnym.

(j.j.)

D. Madej, K. Marasek, K. Kuryłowicz, "Komputery osobiste", Wydawnictwo Komunikacji i Łączności, Warszawa 1987, wydanie I, nakład 150 tys. egz. Cena 500 zł.

ZAMIESZANIE NA EKRANIE

Jednym z bardziej skomplikowanych układów cyfrowych w mikrokomputerach CPC Amstrad jest sterownik monitora ekranowego. odpowiedzialny za kontrole generowanego sygnału wizyjnego.

CRTC 6845 jest programowalny za pośrednictwem 19 wewnętrznych rejestrów, dostępnych przez dwukierunkowy 8-bitowy port, którego tylko dwa adresy (wrócimy do tego nieco później) wystarczą do zmiany zawartości tych rejestrów. Rejestry 16 i 17 odpowiedzialne są za obsługę pióra świetlnego, lecz są niedostępne dla użytkownika, zaś rejestr 18 pośredniczy w programowaniu 18 pierwszych. Interesują nas więc rejestry od 0 do 15.

Rejestr	Funkcja	Zakres
0	Całkowita ilość znaków w wierszu plus synchronizacja	0-255
1	Ilość wyświetlanych znaków w wierszu	0-255
2	Synchronizacja pozioma (pozycja)	0-255
3	Długość synchr. poziomej	0-15
4	Całłkowita ilość wierszy w pionie	0-127
5	Synchronizacja pionowa	0-31
6	Ilość wyświetlanych wierszy	0-127
7	Synchronizacja pionowa (pozycja)	0-127
8	Tryb wyświetlania	0-3
9	Ilość linii w wierszu -1	0-31
10	Linia w wierszu, od której kursor kończy przesłanianie znaku	0-31
11	Linia w wierszu, na której kursor kończy przesłanianie znaku	0-31
12	Adres pamięci ekranu (bajt starszy)	0-255
13	Adres pamięci ekranu (bajt młodszy)	0-255
14	Pozycja kursora (bajt starszy)	0-255
15	Pozycja kursora (bajt młodszy)	0-255

Standardowe wartości wynoszą przykładowo: w MODE 1 R0=63, R1=40, R2=46 i R3 = 142. Parametry ekranu w pionie: R4 = 38, R5=0, R6=25 i R7=30. Do zapisania danych w odpowiednich rejestrach wystarczy znać dwa adresy wspomnianego wyżej PORTu: &BC00 — w celu wybrania rejestru i &BD00 dla wprowadzenia pożądanego parametru. Wpisanie wartości "1" do rejestru 8 wymaga więc wprowadzenia (w BASIC-u) następującej sekwencji: OUT &BC00,8: &BD00,1. Nie zachęcam jednak do daleko idących eksperymentów, gdyż bez dokładnej znajomości wszystkich parametrów technicznych CRTC 6845 i umiejętności programowania można doprowadzić do ZNISZCZENIA układu. Niniejszy opis ma tylko ułatwić zrozumienie potencjalnych możliwości sterownika monitora, natomiast bezpieczne zmiany parametrów zapewni program "PRSCROLL. BIN" utworzony po wpisaniu i uruchomieniu programu ładującego "PRSCROLL.BAS". Program należy wpisywać po wcześniejszym uruchomieniu "Weryfikatora" z poprzedniego numeru "Bajtka". Liczby w nawiasach kwadratowych są

sumą kontrolną generowaną przez "Weryfikator" i służą tylko do sprawdzania poprawności wpisywania linii programu. Należy również pamiętać o wtożeniu dyskietki do stacji przed uruchomieniem programu ładującego. W wyniku załadowania powstałego w ten sposób programu (LOAD "PRSCROLL.BIN", &A000) uzyskujemy dwa sposoby zmiany organizacji wyświetlania ekranu: CAL& A000, ile razy, kierunek — gdzie "ile razy" określa krotność przesunięcia linii całego ekranu, natomiast "kierunek" = 1 dla przesunięcia w górę i = 0 dla przesunięcia w dół; CAL& A000, lewy, prawy, górna, dolna, ile razy, kierunek — gdzie "lewy, prawy, górna, dolna" są parametrami określającymi fragment ekranu (tak jak przy określaniu parametrów WINDOW) a "ile razy, kierunek" mają podobne funkcje jak w poprzednim przykładzie. Krótki program demonstracyjny "DEMO" spowoduje nieco zamieszania wyłącznie dla usprawiedliwienia tytułu publikacji. Za zamieszanie w głowach odpowiada niżej podpisany...

Wojciech Ziółek

```
10 'PRSCROLL (loader PRSCROLL.BIN    [3550]
20 MEMORY &9FFF                       [134]
30 FOR i=&A000 TO 8&A039 STEP 8     [1020]
40 FOR j=i TO i+7:READ  a$            [994]
50 c=VAL("&"+a$):POKE j,c:NEXT:NEXT [3185]
60 SAVE "Prscroll.bin" ,b,&A000,&39 [2607]
70 DATA fe,02,28,26,fe,06,c0,dd      [1302]
80 DATA 5e,04,1d,dd,6e,06,2d,dd       [930]
90 DATA 56,08,15,dd,66,0a,25,dd      [1630]
100 DATA 46,02,c3,e5,d5,af,dd,46      [982]
110 DATA 00,cd,50,bc,d1,e1,c1,10      [805]
120 DATA f1,c9,dd,46,02,c5,af,dd     [1492]
130 DATA 46,00,cd,4d,bc,c1,10,f5     [2026]
140 DATA c9,00,00,00,00,00,00,00     [1564]
150 END                               [110]
```

```
10 'Program demonstracyjny DEMO     [26783]
20 LOAD "prscroll.bin",&A000         [2796]
30 MODE 1:CLS                        [1104]
40 FOR b=1 TO 15:PRINT "CALL
&A000,ilerazy,kierunek":NEXT b       [4542]
50 FOR i=1 TO 4                       [450]
60 CALL &A000,i,0                     [888]
70 FOR a=1 TO 1000:NEXT a            [1143]
80 CALL &A000,i,1                     [891]
90 FOR a=1 TO 1000:NEXT              [1433]
100 NEXT i                            [375]
110 MODE 2:FOR c=1 TO 15             [1950]
120 PRINT "CALL &A000, lewy, prawy, gorna,
dolna, ilerazy, kierunek"            [4931]
130 NEXT c                            [381]
140 FOR i=10 TO 40 STEP 10            [699]
150 CALL &A000,1,i,2,i,3,0           [1359]
160 FOR a=1 TO 500:NEXT a            [1208]
170 CALL &A000,1,i,2,i,3,1           [1358]
180 FOR a=1 TO 500:NEXT a            [1208]
190 NEXT i                            [375]
```

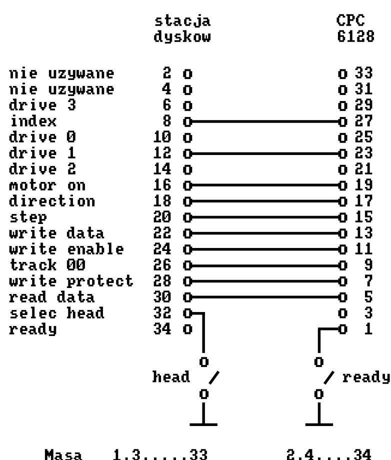
NAPĘD 5 1/4" — DO CPC 6128

Komputery Amstrada CPC 6128 sprzedawane sa z jedna 3 calową stacją dysków. Producent przewidział możliwość dołączenia drugiego takiego samego napędu. Z pewnych względów bardziej użyteczne może się okazać zastosowanie stacji 5 i 1/4 cala.

Do przeróbki potrzebne są dwa 34-stykowe złącza krawędziowe, pół metra 34-żyłowego kabla, dwa przełączniki i dodatkowy zasilacz dwunapięciowy: 12V 700mA, 5V 500mA. Oczywiście niezbędny jest też typowy 5 calowy napęd dysków np, TEAC 360 KB.

Konieczne połączenie między stacją a komputerem przedstawiono na rys.1. Przełączniki "READY" i "HEAD" zapewniają właściwą pracę urządzeń. Sygnał READY podawany ręcznie służy do poinformowania komputera, że w stacji znajduje się dyskietka. Poza programami typu DISCKIT przełącznik ten powinien być stale zwarty. Sygnał "HEAD", także podawany ręcznie, pozwala na dostęp do obu stron dyskietki bez konieczności jej przekładania i robienia dodatkowych dziurek w kopercie.

Jonasz Mayer



Rys. 1 — Połączenie między stacją dysków a komputerem Amstard CPC 6128

1.+12V 700 mA
2. gnd
3. gnd
4.+5V 500 mA

Rys. 2 — Łącze zasilające stacji dysków

AMSTRAD INFO

Jeśli zamierzamy pisać programy, mogące różnie działać w zależności od typu komputera, na którym są uruchamiane, to przedstawiony poniżej program ułatwi nam zrobienie tego. Jeśli nie, to i tak problemy poruszone w tym krótkim artykule mogą okazać się interesujące dla posiadaczy 8-bitowych komputerów firmy Amstrad.

Prezentowany program składa się z dwóch części: jednej napisanej w Turbo-Pascalu i drugiej w assemblerze. Program pascalowy zawiera dwie procedury VERSION I INFO. Każda z nich wypisuje dane o komputerze po wcześniejszym odwołaniu się do systemu operacyjnego poprzez funkcje RSX'a, dostępne dzięki krótkiemu programowi w assemblerze.

W procedurze VERSION rekord INFO ma następujące znaczenie:

a=0 — komputer: CPC 6128
a<>0 — komputer: PCW 8256/8512
b.c — zawierają numer wersji BIOS'a
h,l — zawierają numer wersji ROM'u (istotne tylko dla CPC, ponieważ PCW nie ma żadnego ROM'u. W procedurze INFORMATION rekord ten ma następujące znaczenie:
a=0 — 1 stacja dysków
a<>O — 2 stacje,
b — liczba bloków pamięci, każdy ma 16 kB
c=0 — brak interface'ów, w przypadku CPC oznacza brak tylko RS-232C
c<>0 — oznacza połączony RS-232C dla CPC, a dla PCW oznacza także podłączenie złącza CENTRONICS.

Narzędzia potrzebne do pracy są następujące:
1. Turbo-Pascal — kompilator i edytor,
2. M80 — assembler (jeśli nie mamy, to warto się w niego zaopatrzyć, bo dostępny na dyskietce firmowej RMAC nie akceptuje kodów Z80)
3. GENCOM.COM — umożliwia tworzenie RSX-ów i ich łączenie ze zbiorami typu .COM
4. LINK.COM — konsolidator (ang. LINKER)

Zbiory GENCOM.COM i LINK.COM znaidują się na dyskietkach systemowych.

Metoda postępowania:
1. przy pomocy edytora Turbo-Pascala tworzymy dwa zbiory tekstowe:
a. INF.PAS — wersja źródłowa programu pascalowego,
b. INF.MAC — wersja źródłowa programu assemblerowego.
2. kompilujemy na dysk zbiór INF.PAS wybierając adres końcowy COOO H przy pomocy opcji O i E (patrz opis Turbo-Pascala)!, otrzymujemy zbiór INF.COM
3. na zbiór INF.MAC działamy assemblerem M80:
A>m80 = inf <RETURN>
4. wykonujemy operacje:
A>link inf op
A>ren inf.rsx = inf.prl
A>gencom inf inf

W wyniku poprawnego wykonania tych operacji uzyskujemy zbiór INF.COM, który po uruchomieniu na komputerze da na ekranie wydruk podobny do przedstawionego niżej.

Gotowy program INF nie ma oczywiście większego znaczenia poza prostą demonstracją, ale jeśli dołączymy procedury pascalowe do własnego programu napisanego w tym języku i wykonamy podobne operacje do opisanych wyżej, to uzyskamy własny program, który może korzystać z istotnych informacji systemowych. Jedyną zmianą jest zastosowanie operacji:

A>gencom WŁASNY inf jeśli nasz program pascalowy nazywa się WŁASNY.PAS i po skompilowaniu dostajemy zbiór WLASNY.COM

Opisana metoda może być przeniesiona, także na programy w języku BASIC, ale jest to bardziej kłopotliwe, co jest kolejnym argumentem w ogólnej dyskusji, jakiego używać języka programowania.

Przedstawiony zestaw narzędzi i metoda postępowania stanowi idealne rozwiązanie przy tworzeniu własnych RSX'ów.

Informacja o komputerze AMSTRAD PCW 8256/8512 BIOS version: 1,4
Konfiguracja.
1. Pamięć RAM 512 kB
2. 1 stacja dysków
3. brak interfejsów Przykładowy wydruk programu:

Jonasz Mayer

LISTING 1

```
program info_version;
(**********************************************************)
(* program dla komputerow 8-bitowych firny Amstrad       *)
(* pozwalajacy okreslic komputer i jego konfiguracje     *)
(*               (C)  J. I                               *)
(*             grudzien 1987                             *)
(* uwaga: korzysta z funkcji RSX'a o numerze 69          *)
(**********************************************************)
const
    cdvers = $00E3;
    cdinfo = $00E6;
    rsxnr  = 69;      (* 45H *)
type
    HexStr              = string(.4.);
    info = record
                          a   : byte;
                          b,c : byte;
                          h,l : byte;
                    end;
var
pinf   : ^info;   PCW : boolean;
procedure Version;
begin
    pinf : = ptr (Bdoshl(rsxnr,cdvers));      with pinf^
    do begin
            writeln;   PCW := false; if a=0
then write(' AMSTRAD CPC 6128    ')
else begin
          write(' AMSTRAD PCW 8256/8512 ');    PCW := true;
end;
write(' BIOS version: ',b:l,'.',c:l);
if not PCW
then write('   ROM version: ',h:l,'.',l:l);
end;
writeln;
end;  (* yersion *)

procedure information;
begin
    pinf ;= ptr (BdosHL (rsxnr,cdinfo));
    with pinf^ do begin
       writeln;
       writelnl' Konfiguracja;');
       writeln(' l. Pamiec RAM ',b*16:4,' kB');
if a=0
then writeln(' 2. 1 stacja dyskow')
else writeln(' 2, 2 stacje dyskow');
if c=0
then if PCW
then writeln(' 3. brak interfejsow')
else writeln(' 3. brak RS-232C')
else if PCW
then writeln(' 3. Centronics i RS-232C')
else writeln(' 3. RS-232C');
end;
end; (* of information *)

begin
writeln ('Informacja o komputerze ':40);
version:
information;
end.

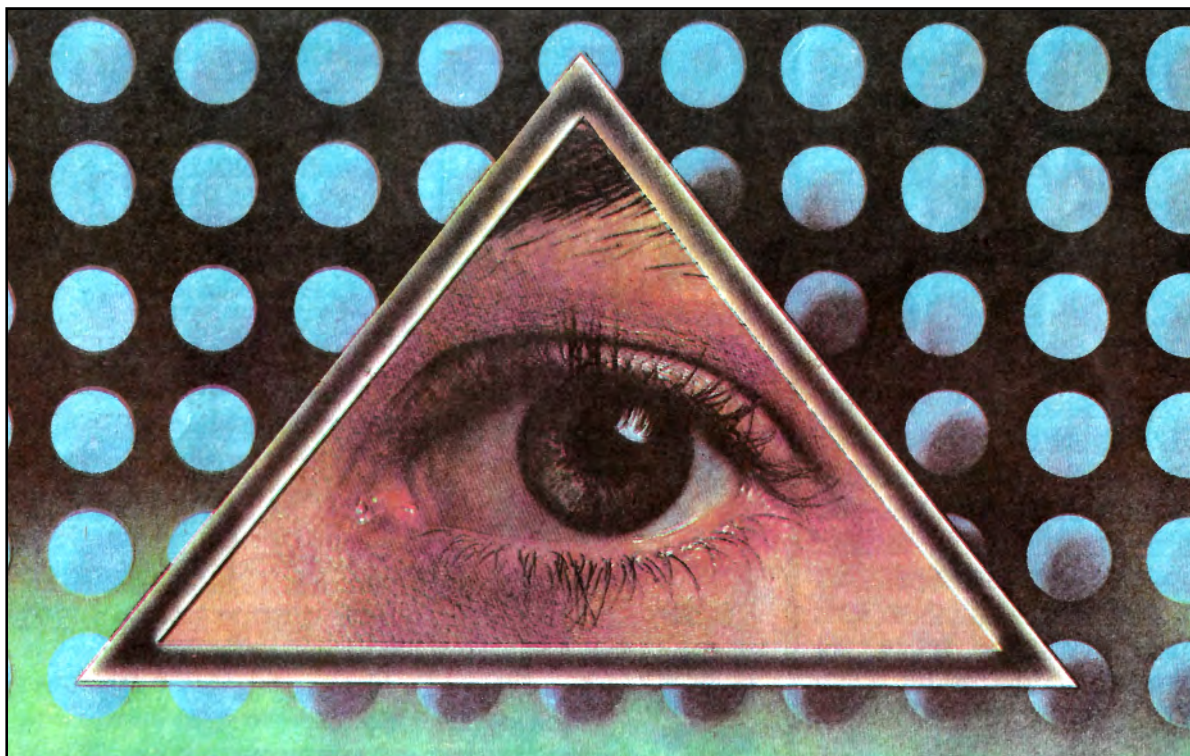

zbior INF.PAS
```

LISTING 2

```
; ****************************************************
; * rsx dostepu do informacji o sprzecie              *
; *          (C) J. M.   grudzien 1987               *
; * dane wejsciowe:                                  *
; *     rejestr DE zawiera nuaer procedury systemowej *
; *      1. E6 - cd_info                             *
; *      2. E3 - cd_version                          *
; ****************************************************
        .z80
 rsxnr  equ 69        ; 45H
 cdinfo equ 00E6H
 cdvers equ 00E3H
 warm   equ 0001H
 userf  equ 3*(30-1)
;
      cseq
; rsx header
     ds     6
     jp     start next:jp     0000h
     dw     0
     db     0FFH
     db     0
     db     'infovers'
     db     0,0,0
start:
     push   af
     ld     a,c
     cp     rsxnr
     jp     z,begin
     pop    af
     jp     next
begin:
     ld     (func),de
     ld     hl,(warm)
     ld     bc,userf
     add    hl,bc
     call   jphl
func:dw     00E6H ; E6 info, E3 vers
     ex     de,hl
     ld     hl,buffer
     ld     (hl),a
     inc    hl
     ld     (hl),b
     inc    hl
     ld     (hl),c
     inc    hl
     ld     (hl),e
     inc    hl
     ld     (hl),d
     ld     hl,buffer
     pop    af
     ret
jphl:jp     (hl)
buffer:
     ds   5
     end

;
;   zawartosc zbioru  INF.MAC
```

System operacyjny czyli program nadzorujący prace komputera. Już dwa miesiące śledziliśmy jego rozwój i dziś kontynuujemy to zajęcie. Pamiętamy, że rozwój oprogramowania systemowego był wymuszany przez lenistwo obsługi oraz burzliwy rozwój możliwości sprzętu.



SYSTEM OPERACYJNY

Cz. III

Pożegnaliśmy się w sytuacji dość niejasnej. Procesor miał czytać dane nie bezpośrednio z czytnika kart, lecz za pośrednictwem znacznie szybszych pamięci taśmowych czy dyskowych*). Kto w takim razie ma je wczytać do tych pamięci? Jedno z możliwych rozwiązań było następujące: malutki, tani komputerek wczytuje paczkę kart perforowanych, stanowiących kompletne zadanie**) na taśmę magnetyczną. Po niej następną i następną, tworząc w ten sposób na taśmie tzw. wsad zadań, który następnie przenoszono na duży komputer, ten zaś zadania po kolei realizował, a wyniki w podobny sposób (za pośrednictwem taśmy) wracały do wydrukowania na drukarce. System operacyjny dużego komputera nadzorował przejście od jednego zadania do drugiego.

Rozwiązanie to bardzo zwiększało wykorzystanie drogiego, szybkiego procesora oraz urządzeń wejścia/wyjścia, kosztem niewielkiej rozbudowy sprzętu. I tu zaskakujący zwrot sytuacji: drugim składnikiem kosztu była ogromna niedogodność tego systemu przetwarzania dla człowieka — separacja od komputera, długie oczekiwanie na wyniki, skomplikowane zabiegi z noszeniem mnóstwa taśm. Bardzo szybko okazało się, że takiego kosztu nikt nie chce ponosić. Musiało więc powstać lepsze rozwiązanie, ale o nim za chwilę, gdyż będzie nam łatwiej je zrozumieć gdy przejdziemy przez następny etap rozwoju. Czytelnik, który zechciał poświęcić odrobinę uwagi uprzednim odcinkom, z łatwością zgadnie, że procesory komputerów znów przyśpieszyły.

SZYBKI PROCESOR — WOLNY DYSK

W związku z tym dyski stały się stosunkowo wolne i w oczekiwaniu na transmisję danych procesor wylegiwał się bezczynnie np. przez całą jedną dziesiątą każdej sekundy (jeżeli się robi kilka milionów operacji na sekundę, to jest to naprawdę poważna przerwa).

Rozwiązanie jest następujące: trzeba umieścić w pamięci operacyjnej komputera kilka programów na raz (oczywiście w oddzielnych obszarach). Jeśli jeden z nich zażąda przesłania danych z dysku, to procesor nie czeka aż to nastąpi, tylko zaczyna wykonywać inny program, wybrany spośród czekających w pamięci. Genialnie proste, wejdźmy jednak w skórę tego, który ma tę wieloprogramowość zrealizować w praktyce.

Problemów jest mnóstwo: skoro w momencie żądania transmisji procesor ma zacząć liczyć inny program, to potrzebny jest ktoś, kto tę transmisję zrealizuje i umieści dane z dysku w pamięci operacyjnej. Czyli pojawiają się nowe elementy architektury komputera: wyspecjalizowane urządzenia do sterowania wejściem/wyjściem, noszące nazwę kanałów; dalej bloki DMA (Direct Memory Access — bezpośredniego dostępu do pamięci), pozwalające kanałom pisać i czytać pamięć bez udziału procesora. Wreszcie, gdy transmisja już została zakończona trzeba procesor zawiadomić — a więc rozbudowany system impulsów przerywających normalny tok pracy. No i jeszcze "drobiazg" — jeśli w ciągu jednej sekundy procesor przełącza się na obsługę kilkunastu różnych programów, to na pewno przełączaniem tym nie steruje człowiek! A kto? SYSTEM OPERACYJNY. Napisałem go drukowanymi literami, bo teraz to naprawdę ma co robić

Oczywiście małe komputerki przepisujące karty perforowane na taśmę znikają automatycznie. Ich funkcje wykonują kanały. Tak więc komputer ma z powrotem swoje urządzenia wejścia/wyjścia, ale podłączone nie bezpośrednio do procesora, ich pracą sterują kanały. Skoro umiemy już godzić wczytywanie danych z wolnych czytników z efektywnym wykorzystaniem procesora, to możemy pójść jeszcze dalej i dołączyć do naszego systemu cyfrowego***) końcówki, składające się z klawiatury i monitora, pozwalające człowiekowi kontaktować się z maszyną bez pośrednictwa kart czy taśmy perforowanej. Dzięki swej ogromnej szybkości jeden procesor może obsługiwać równocześnie wielu użytkowników (wiele końcówek) i każdemu z nich wydaje się, że ma cały komputer dla siebie. Taki sposób eksploatacji nazwano wielodostępem. Typową praktyką jest również instalowanie końcówek zdalnych — nie stojących bezpośrednio przy komputerze, lecz np. w innej miejscowości i współpracujących za pośrednictwem łączy telekomunikacyjnych.

DUŻE JEST SKOMPLIKOWANE

Wszystkie elementy muszą współdziałać w efektywny i harmonijny sposób, co nakłada na system operacyjny ogromne zadania. Musi on rejestrować wszystkie wpływające zadania, ustalać kolejność w jakiej będą wykonywane, jeśli zadanie wczytane z kart nie może być wykonane od razu to przechować je na dysku, rozsyłać wyniki na odpowiednie drukarki, zarządzać pamięcią składającą się z wielu stacji dysków i taśm, a wszystko to z absolutną precyzją — przekłamanie jednego adresu transmisji może zdezorganizować cały system. Trzeba też dopilnować, aby żadne zadanie nie czekało zbyt długo, udostępniać wszystkim użytkownikom programy publiczne (np. edytory, translatory, bazy danych itd.), wreszcie za wszystkie wykonane operacje obliczać i wystawiać rachunki użytkownikom.

Dodatkowo dochodzą problemy ochrony użytkowników przed sobą (w tej samej pamięci siedzą obok siebie programy różnych ludzi a każdy ma swoje tajemnice), oraz systemu przed użytkownikami. Łatwo sobie wyobrazić pełną dezorganizację systemu, spowodowaną przez użytkownika, który dla eksperymentu zastąpił jeden z systemowych podprogramów obsługi zbiorów, własnym podprogramem, i (np. przez pomyłkę) w miejscu, w którym była zapisana kartoteka zbiorów, zapisał list do narzeczonej.

Aby taka ochrona była skuteczna potrzebne są nowe mechanizmy sprzętowe, np. pamięć operacyjna, która automatycznie blokuje próbę zapisu lub odczytu poza przydzielonym dla jednego programu obszarem.

CO MOŻE MIKRO?

Oczywiście oprócz ogromnych systemów cyfrowych budowano (i buduje się) także mniejsze, w których część z opisanych elementów w ogóle nie występuje. Jednak ich konfiguracje są ciągle o wiele bardziej rozbudowane niż przeciętny mikrokomputer, szczególnie domowy.

Mikrokomputery nie powstały jako zminiaturyzowane wersje dużych systemów cyfrowych****), nie odziedziczyły ani ich pełnej architektury ani systemów operacyjnych. Początkowo w ogóle nie miały żadnego systemu operacyjnego, także programowanie odbywało się w języku maszynowym...

Jeśli ktoś zrobił mi ten zaszczyt i przeczytał uważnie dwa poprzednie odcinki, to pamięta, że tak właśnie zaczynała się historia systemów operacyjnych dużych komputerów. Historia, którą opowiadałem Wam nie przypadkiem, gdyż ona teraz, na naszych oczach, powtarza się dla sprzętu mikro!

Na prostych, zwykle nie posiadających pamięci dyskowych, maszynkach domowych polecenia sterujące pracą maszyny wmontowane są w BASIC (komendy RUN, SAVE itp.).

Ogromną popularność zdobył system CP/M, który nie tylko jest nadzorcą maszyny, lecz organizuje na dyskietce własny, prosty system plików, oraz udostępnia programom użytkowym zestaw systemowych procedur, obejmujących wejście/wyjście.

Wraz z rozpowszechnieniem się sprzętu mikro i rozwojem jego możliwości wzrastają apetyty użytkowników i oczywiście rozwijają się również systemy operacyjne. Mogą one obsługiwać równocześnie kilka urządzeń. Np. w systemie DOS*****) dla IBM PC można zlecić wydruk zawartości zbioru dyskowego i zająć się

innym zadaniem — drukowanie odbywa się równolegle z przetwarzaniem następnego zadania.

Równocześnie pojawiają się liczne próby realizacji w oparciu o mikrokomputery systemów wieloprogramowych i wielodostępnych. Nie zawsze są one udane — moc obliczeniowa procesora jest czasami niewystarczająca, brakuje odpowiednio rozbudowanego systemu przerwań i autonomicznego wejścia/wyjścia, dynamicznego zarządzania pamięcią operacyjną. Jednak sam fakt podejmowania takich prób (oczywiście w połączeniu z ciągłym postępem technologii — "komputery przyspieszają") skłania producentów do rozbudowy konfiguracji nowych modeli o elementy ułatwiające np. wieloprogramowość.

Istnieją już systemy wielodostępne dla mikrokomputerów 16-bitowych, najbardziej znane z nich to XENIX i UNIX, będące przeniesieniem bardzo popularnego na minikomputerach systemu UNIX. Mogą one działać np. na IBM PC AT i zaczynają zdobywać popularność także i w Polsce.

Czego możemy spodziewać się w niedalekiej przyszłości, czy rozwój sprzętu i oprogramowania w skali mikro jest i będzie wierną kopią rozwoju dużych maszyn? Myślę, że nie. Historia powtarza się, ale nigdy dosłownie. Przecież konstruktorzy i informatycy znają poprzednie rozwiązania i wiedzą, które dobrze sprawdzały się w praktyce, a które warto wyeliminować. Z drugiej strony ciągły rozwój elektroniki może w każdej chwili otworzyć nowe, dotychczas nieznane możliwości.

Inny aspekt to odbiorcy sprzętu. Poprzednio komputery były programowane i obsługiwane przez stosunkowo nieliczną grupę fachowców — przeciętny klient stawiał zadanie i otrzymywał wyniki. Teraz nastąpiło znaczne umasowienie, każdy sam obsługuje swój komputer osobisty. Stawia to nowe wymagania przed oprogramowaniem. Musi ono być łatwe w obsłudze nawet dla laika, stąd burzliwy rozwój nowych koncepcji komunikacji operatora z programem, wymyślne menu, ikony itd.

Generalnie jednak droga do osiągnięcia przez mikro pełnych możliwości dużych instalacji jest jeszcze daleka, dlatego warto zachować trochę dystansu do twierdzenia: "komputery wkroczyły w nasze życie bo pojawiło się Spectrum". Mnie osobiście inwazja sprzętu mikro uświadamia gdzie nie mamy komputerów, chociaż powinniśmy mieć i to już dawno!

Na koniec wreszcie chcę wyrazić nadzieję, iż przekonaliśmy się wspólnie o tym jak wiele zadań wykonuje za nas system operacyjny, a co za tym idzie o jego wartości i niezbędności, a także o tym, że zrobienie dobrego systemu operacyjnego to duża sztuka.

*) Mamy tu przykład buforowania: szybka pamięć magnetyczna stanowi bufor między wolnym czytnikiem a bardzo szybkim procesorem.

**) Treść programu, dane itd.

***) Przy tak rozbudowanych instalacjach bardzo często zamiast komputer mówimy właśnie system cyfrowy.

****) Ich przodków należy szukać wśród elektronicznych kalkulatorów.

*****) Uwaga na nazwę DOS. Jest to angielski skrót terminu Disk Operating System (dyskowy system operacyjny) i wielu autorów używa go jako nazwy swojego systemu. Stąd DOS można spotkać na wielu różnych typach komputerów, oczywiście oznacza to za każdym razem zupełnie inny system.

Andrzej Pilaszek

BUFORY

Zanim zajmiemy się informatyką przypomnijmy sobie dokładne znaczenie słowa bufor w zastosowaniach tradycyjnych. Z grubsza rzecz biorąc jest to urządzenie łagodzące wstrząsy przy zetknięciu dwóch pojazdów poruszających się z różnymi prędkościami. Zaraz zobaczymy czy przechodząc do informatyki słowo to zachowało podstawowe znaczenie.

Przyjrzyjmy się współpracy procesora z klawiaturą. Procesor wykonuje rozkaz "Czytaj znak z klawiatury"*). Jeśli jest wciśnięty jakiś klawisz, to odpowiedni kod zostanie wczytany, czyli umieszczony w umówionym miejscu pamięci komputera. Teraz drobna dygresja: kiepski program po wykonaniu rozkazu czytania idzie dalej i po prostu przetwarza pobrany (?) znak. Dobry sprawdzi, czy rzeczywiście coś zostało wczytane, bo może żaden klawisz nie był wciśnięty. W tym wypadku zaczyna działać w pętli — wykonuje wczytanie i sprawdzenie poprawności aż do momentu faktycznego wczytania znaku. Unika w ten sposób przetwarzania śmieci. Takie sprawdzanie poprawności wykonania operacji jest niezbędne gdy sami programujemy obsługę urządzenia (np. w języku wewnętrznym maszyny).

Jeśli natomiast komunikujemy się z nim za pośrednictwem procedur wejścia/wyjścia dostarczonych przez język programowania czy system operacyjny, to problemy te są już załatwione. Np. instrukcja INPUT w BASIC-u czeka aż operator wpisze dane — nie musimy w naszym programie umieszczać pętli oczekiwania na pojedyncze znaki.

Wróćmy do wczytywania: klawisz wciśnięty, znak wczytany. Jeśli jednak, po wczytaniu każdego znaku program musi wykonać dłuższe rachunki, to może się okazać, że operator pisze szybciej niż komputer czyta i zanim jeden znak zostanie wczytany naciśnie już następny. Oczywiście część znaków ginie. Chyba że wyposażymy klawiaturę w niewielką pamięć, w której będą zapisywane naciskane klawisze. Procesor komputera będzie pobierał znaki nie bezpośrednio z klawiatury, lecz z tej pamięci, czyli właśnie BUFORA. Pamięć buforowa służy więc do amortyzacji różnicy prędkości między pisaniem a wczytywaniem znaków (w omówionym przypadku różnica na korzyść operatora). W przypadku klawiatury typowe długości bufora to 8 lub 16 znaków.

KONIEC GUBIENIA ZNAKÓW

Oczywiście w dalszym ciągu istnieje możliwość przepełnienia bufora, np. jeśli program liczy przez godzinę i nic nie wczytuje. Dlatego niezbędny jest układ sygnalizujący brak miejsca na nowe znaki i blokujący wprowadzanie. (Zwykle w takiej sytuacji komputer reaguje piskiem na każde naciśnięcie klawisza). UWAGA: nie należy mylić zawartości bufora klawiatury, ze znakami, które już są napisane i odbite na ekranie, lecz jeszcze nie zostały przesłane do przetwarzania naciśnięciem ENTER (lub RETURN). Znaki siedzące w buforze klawiatury nie zostały jeszcze zauważone przez resztę komputera, "wyskoczą na ekran" gdy procesor przestanie być zajęty i poświęci im uwagę.

Wniosek praktyczny jest następujący: jeśli pracujemy z programem konwersacyjnym, który sterujemy np. naciskając spację, i program, obliczając coś, przestał reagować na klawiaturę, to nie warto wciskać tej spacji wielokrotnie. Pierwsza już siedzi w buforze klawiatury i czeka na pobranie. Następne też tam trafiają i będą po kolei pobierane i interpretowane jako następne polecenia, co zwykle wyśmienicie wyprowadza program w maliny.

WĄSKIE GARDŁO – WOLNA DRUKARKA

Popatrzmy teraz co się dzieje na wyjściu, czyli na współpracę komputer — drukarka. Komputer jest w stanie produkować wyniki dużo szybciej niż drukarka przenosić je na papier. Często jednak produkcja tych wyników jest nierównomiernie rozłożona w czasie. Rozpatrzmy następujący przykład: pewien program rozwiązuje 10 równań. Obliczenia dla jednego równania trwają 100 sekund, a po ich zakończeniu wysyła na drukarkę 10 linii wydruku (czas przesłania komputer — drukarka będziemy systematycznie pomijać). Wydruk jednej linii zajmuje drukarce 5 sekund.

Po otrzymaniu znaku drukarka wysyła komputerowi sygnał: "Brak gotowości do przyjmowania znaków", czym zmusza go do oczekiwania. Następny znak można wysłać dopiero gdy drukarka napisze poprzednio wysłany i wyśle komputerowi sygnał: "Gotowe, wysyłaj". Przy tym schemacie obliczenia i wydruk dla jednego zestawu zajmują: 100s + 10linii*5s = 150s a dla dziesięciu zestawów 1500s.

Jeśli jednak możemy wysyłać na raz całą linię (drukarka ma bufor na tyle znaków ile mieści się w linii), a następną linię gdy tylko skończy się druk poprzedniej, to komputer musi czekać na drukarkę tylko do momentu wysłania ostatniej linii, czyli podczas drukowania dziewięciu, razem 9*5s = 45s, a czas wykonywania obliczeń wynosi 10*(100s + + 45s) = 1450s, a więc nieco lepiej.

Zwróćmy uwagę, że bufor został wprowadzony jakby między urządzenie bezpośrednio drukujące a komputer. Komputer wpisuje swoje dane do bufora z jednej strony, a z drugiej strony tegoż bufora są one pobierane do druku.

Co się stanie, jeśli powiększymy pojemność bufora drukarki do pełnych dziesięciu linii? Spróbujcie się najpierw zastanowić sami. Gotowe? To jedziemy. Po zakończeniu obliczeń dla jednego zestawu, komputer zaczyna przesyłać wyniki na drukarkę. Dopóki bufor nie będzie pełen drukarka przyjmuje je bez problemów, czyli (w bardzo krótkim czasie, któ ry pomijamy) wszystkie dane zostały wyprowadzone. Przynajmniej z punktu widzenia procesora, który natychmiast przystępuje do następnych rachunków. W tym czasie drukarka, pracując równolegle z procesorem, przenosi zawartość bufora na papier. Czas wykonywania obliczeń: 10* 100s = 1000s. Po tym czasie komputer może rozpocząć następne zadania. Na koniec drukowania musimy poczekać jeszcze dodatkowe 50s.

Efekt ostateczny: poprawa efektywności wykorzystania komputera o prawie jedną trzecią, a więc bardzo znaczna.

Koszty: wyposażenie drukarki w dodatkową pamięć, wielkości rzędu kilku kilobajtów oraz układy obsługi bufora. Przy dzisiejszych cenach pamięci i elektroniki są to koszty nieznaczne.

Myślę, że już nikt nie ma wątpliwości: jak to jest możliwe? Program już się skończył, widać to na monitorze, a drukarka jeszcze pracuje.

TROCHĘ DODATKOWYCH KŁOPOTÓW

Rola bufora drukarki jest jednak bardziej złożona. Bywa bowiem całkiem odwrotnie — komputer generuje po kilka znaków, w sporych odstępach czasu. Mogą one być gromadzone w buforze drukarki i wyprowadzane na papier dopiero gdy linia zostanie zapełniona, lub komputer prześle bezwzględny rozkaz: "Koniec linii". Właściwie nie ma to dla nas znaczenia, ale czasami zdarza się tak, że program wyśle część znaków na drukarkę, lecz zanim zapełni lub zakończy linię, jego realizacja zostaje przerwana, np. przez wystąpienie błędu. Wtedy część znaków "znika", wydaje się, że działanie programu zostało przerwane nie tam, gdzie faktycznie to nastąpiło i szukamy błędu nie tam gdzie jest naprawdę. Piszę o znikaniu w cudzysłowie, gdyż tak naprawdę, to te znaki są. Gdzie? Oczywiście w buforze drukarki. Naprawdę mogą zniknąć dopiero gdy wyłączymy ją z sieci. Jeśli nie, to brakujące znaki pojawią się na początku następnego wydruku.

Podobna sytuacja może wystąpić jeśli po ostatniej instrukcji pisania w programie nie wyprowadzamy znaku końca linii. Może ale nie musi — dobry kompilator lub interpreter po napotkaniu instrukcji kończącej program, może spowodować opróżnianie buforów wszystkich urządzeń.

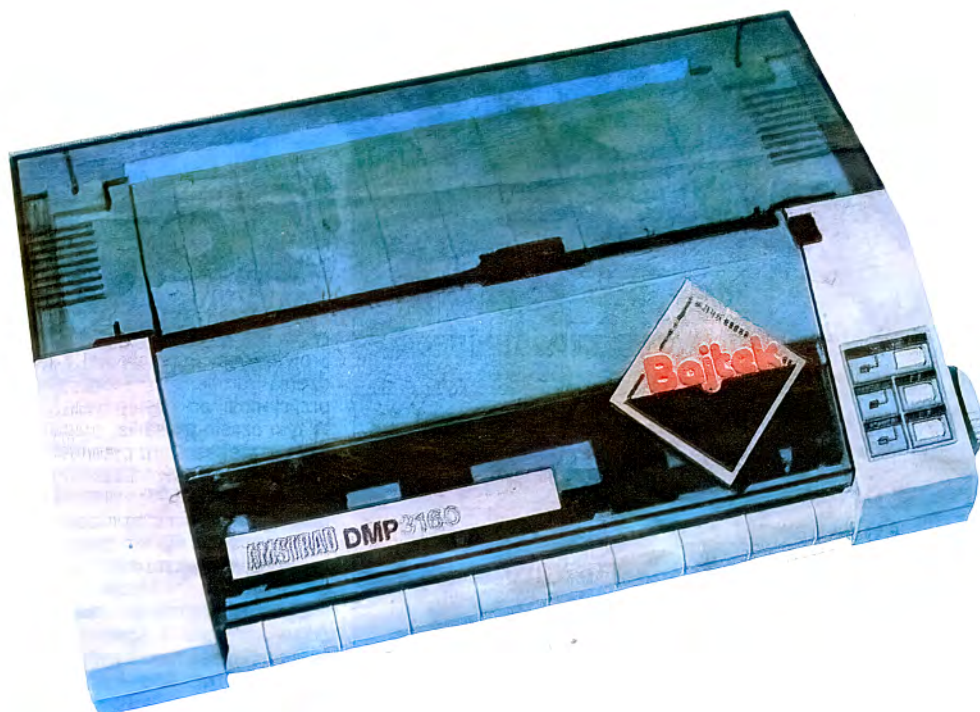
Buforowanie nie ogranicza się tylko do drukarki i klawiatury. Wprost przeciwnie, jest to technika bardzo rozpowszechniona. Stosujemy ją wszędzie tam, gdzie współpracują ze sobą urządzenia o różnych prędkościach działania, aby zoptymalizować wykorzystanie urządzeń, zmniejszyć czas wykonywania obliczeń i wreszcie zapobiec gubieniu informacji, gdy urządzenie wolniejsze nie nadąża i nie jest w stanie zatrzymać pracy szybszego.

*) Oczywiście jest to umowne uproszczenie, zwykle takiego rozkazu nie ma, a przeczytanie znaku wymaga wykonania ciągu kilku rozkazów procesora. Pożegnaliśmy się w sytuacji dość niejasnej.

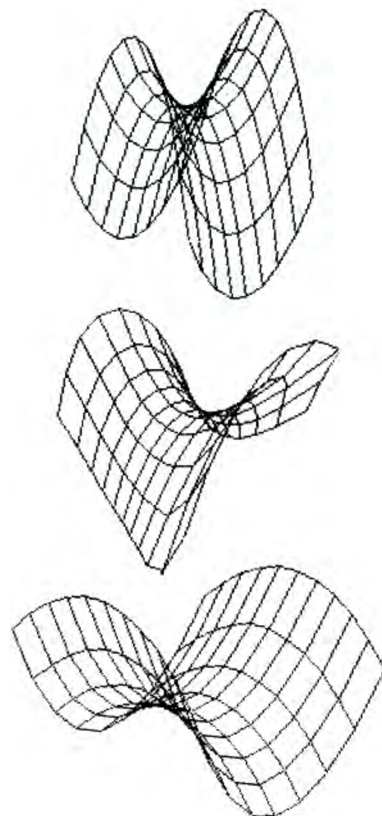
Andrzej Pilaszek

TEST





Drukarkę Amstrad DMP3160 otrzymaliśmy do testowania dzięki uprzejmości pana Andrzeja Łukomskiego, właściciela firmy wysyłkowej Polanglia będącej jedynym oficjalnym przedstawicielem Amstrada w Polsce.



DRUKARKA AMSTRAD DMP 3160

DMP3160 jest 9-igłowa drukarka mozaikowa. Jest ona przeznaczona zasadniczo do pracy z komputerami Amstrad PC oraz innymi komputerami kompatybilnymi z IBM PC. Może także współpracować z Amstradami serii CPC. a przy użyciu interface'u Centronics również z innymi komputerami domowymi (Atari XL/XE. Commodore 64/128. Spectrum 48/+/+2/+31.

BUDOWA I DZIAŁANIE

Już na pierwszy rzut oka odróżnia się od innych drukarek nietypową konstrukcją. Papier jest podawany od przedniej strony i przesuwany poziomo bez opasywania watka, a więc odwrotnie niż w najczęściej dotychczas spotykanych modelach. Umożliwia to drukowanie nawet na dość grubym papierze o niewielkich wymiarach (np. koperty pocztowe), co normalnie jest raczej kłopotliwe. Przy korzystaniu z papieru perforowanego w bloku lub z papieru w rolce można drukarkę unieść do góry na specjalnych, składanych nóżkach i papier umieścić pod nią. Wałek o standardowej długości jest wyposażony w prowadnicę papieru i umożliwia użycie papieru o szerokości od 100 do 255 mm. Oczywiście dopuszczalne są wszystkie formy papieru: perforowany, z rolki lub w arkuszach. Jednorazowo można drukować oryginał z dwiema kopiami.

Sposób prowadzenia papieru wymusił pionowe ustawienie głowicy drukującej. Jest ona przesuwana po prowadnicy przez silnik krokowy za pośrednictwem paska zębatego. Taśma jest umieszczona w bardzo nietypowej, dwuczęściowej kasecie w taki sposób, że przebiega nad i pod głowicą drukującą i jest naciągana sprężyną umieszczoną w jednej z połówek kasety. Przesuw taśmy wymusza jednokierunkowy zaczep na obudowie głowicy. Samo drukowanie odbywa się poza wałkiem na płaskiej podkładce z twardego tworzywa sztucznego. Druk jest możliwy w trybie normalnym (draft) i korespondencyjnym (NLQ). Prędkość druku wynosi 160 znaków na sekundę (character per second — CPS), a w trybie NLQ — 40 CPS. W poprzednim modelu — Amstrad DMP3000 — prędkości te wynosiły odpowiednio 105 i 26 CPS.

Znaki drukowane są z matrycy 9 na 9 punktów. Możliwe jest użycie czterech podstawowych krojów pisma: Pica — 10 znaków na cal (character per inch — CPI), Elitę — 12 CPI, Condensed — 17 CPI oraz Proportional — liczba znaków zależy od ich rodzaju. Daje to odpowiednio 80, 96 i 132 znaki w linii. Krój Proportional jest także możliwy do uzyskania w trybie NLQ. Dodatkowo każdy z krojów pisma może być drukowany w podwójnej szerokości (5, 6 i 8,5 CPI). W trybie draft wszystkie kroje pisma mogą być pochylone (italics). Można wybrać odstęp między wierszami 1/6, 1/8 lub 7/72 cala, albo zaprogramować odstęp n/72 lub n/216 cala.

Drukarka jest przystosowana do pracy w dwóch standardach: Epson FX i IBM Graphics Printer. Wyboru standardu dokonuje się przełącznikiem — programowe przełączenie jest niemożliwe. W trybie Epson drukarka dysponuje zestawem 96 znaków oraz ośmioma zestawami międzynarodowymi po 12 znaków. Znaki o kodach powyżej 127 są wybierane z zestawów podstawowych i drukowane kursywą. Oprócz tego możliwe jest zaprogramowanie przez użytkownika własnego zestawu znaków. W trybie IBM dostępny jest pełny zestaw znaków graficznych standardu IBM (razem 226 znaków). Niemożliwe jest jednak korzystanie z zestawów międzynarodowych i programowanie własnych znaków.

Sterowanie pracą drukarki odbywa się za pomocą przycisków znajdujących się na przedniej ściance. Dostępne są następujące funkcje: połączenie lub odłączenie drukarki i komputera (ON LINE), przesunięcie papieru o stronę (FF — Form Feed) oraz o wiersz (LF — Line Feed). Dodatkowo włączenie drukarki z wciśniętym przyciskiem LF wywołuje program testujący, a równoczesne wciśnięcie LF i FF przy włączaniu drukarki ustawia ją w trybie kopiowania szesnastkowego. Obok przycisków umieszczone są trzy diody LED sygnalizujące włączenie zasilania, połączenie z komputerem i brak papieru. Na tylnej ścianie, w pobliżu złącza Centronics znajduje się zespół 18 mikroprzełączników do trwałego ustalania parametrów pracy drukarki.

EKSPLOATACJA

Drukarka wykorzystywana była do współpracy z różnymi komputerami: IBM PC, Amstrad CPC-6128 i Atari 800XL. Ze wszystkimi komputerami drukarka pracowała poprawnie. Dzięki zgodności z drukarkami Epson FX możliwe jest wykorzystanie "driver'ów" przeznaczonych dla tego standardu. Pozwala to na wydruk wszystkich krojów pisma wraz z dodatkowymi elementami — podkreślanie, wytłuszczanie, pisanie indeksów i wykładników. Jakość pisma jest dobra i przewyższa drukarki Star, ustępując jednak drukarce Triton (zob. "Bajtek" 12/87), która daje wyjątkowo wyraźny wydruk. Także w trybie graficznym drukarka pracowała poprawnie, choć pozostawiała widoczny ślad przejścia głowicy w miejscach bardziej zaczernionych. Przykłady wydruków wykonanych na DMP3160 pokazane są obok.

Test Bajtka - drukarka Amstrad DMP3160
Test Bajtka - drukarka Amstrad DMP3160
Test Bajtka - drukarka Amstrad DMP3160
Test Bajtka - drukarka Amstrad DMP3160
Test Bajtka - drukarka Amstrad DMP3160



Użytkowanie drukarki jest łatwe i wygodne. Przy korzystaniu z różnych rodzajów i formatów papieru ustalenie jego położenia w drukarce jest znacznie ułatwione przez oznaczenia formatów naniesione ponad miejscem wprowadzania papieru. Samo wprowadzanie papieru jest bardzo proste z powodu jego poziomego przechodzenia przez drukarkę. Jednak przy korzystaniu z papieru ciągłego (rolka lub blok) ujawnia się dość uciążliwa wada. DMP3160 w odróżnieniu od drukarek innych firm nie posiada krawędzi umożliwiającej oderwanie zapisanej części papieru.

Przy stosowanym w naszej redakcji systemie drukowania jednego egzemplarza w trybie korespondencyjnym i kopii w trybie draft odczuwalny jest brak na przedniej ściance drukarki przycisku przełączającego te tryby (NLQ — draft). Przycisk taki jest już standardem w drukarkach firmy Star i wielu innych.

Ostatnio daje się zauważyć interesująca tendencja. Każda nowa drukarka — nawet jednej firmy — pojawiająca się na rynku ma inny system prowadzenia taśmy i inną kasetę do taśmy. W DMP3160 zastosowano taśmę o szerokości 6 mm i długości około 70 cm umieszczoną w wyjątkowo nietypowej dwuczęściowej kasecie. Rodzi to obawy, że taśma będzie wymagała częstego nasączania, a ewentualna wymiana może nastręczać wiele kłopotów.

Praca drukarki jest dość cicha, głównie dzięki osłonięciu głowicy drukującej od strony użytkownika innymi elementami mechanizmów i ich obudową.

Obudowę drukarki wykonano z jasnego tworzywa, a pokrywa jest przyciemniana. Wykonanie jest dokładne, a przyciski i przełączniki pewne w działaniu. Całość urządzenia prezentuje się nowocześnie, choć nie ekstrawagancko. Pewien niepokój może budzić delikatność drukarki, lecz jest to tylko wrażenie spowodowane nietypową konstrukcją.

PODSUMOWANIE

Drukarka Amstrad DMP3160 nie jest jakąś rewelacją, ale stosunkowo solidnym urządzeniem o przeciętnym standardzie. Posiada wszystkie możliwości drukarek średniej klasy i dzięki dwóm trybom pracy (Epson i IBM) znajdzie niewątpliwie szeroki krąg nabywców. Ze względu na dość krótką i nietypową taśmę nie polecamy jej jednak do pracy biurowej, w której wymagana jest duża liczba wydruków.

Zalety drukarki:
- proste prowadzenie papieru z wprowadzaniem z przodu
- duża liczba krojów i formatów pisma
- dwa tryby pracy: Epson i IBM
- możliwość definiowania polskich liter
- duża szybkość druku
- łatwe i wygodne wprowadzanie różnych formatów papieru

Wady drukarki:
- nietypowa kaseta na taśmę
- brak krawędzi do odrywania zadrukowanej części papieru z rolki
- brak przełącznika trybów NLQ i draft na przedniej ściance

Wojciech Zientara

PARAMETRY TECHNICZNE DRUKARKI DMP3160 WG SPECYFIKACJI PRODUCENTA

głowica	**— 9-igłowa, możliwość wymiany przez użytkownika**
szybkość druku	**— 160 znaków na sekundę w trybie zwykłym (draft) 40 znaków na sekundę w trybie korespondencyjnym (NLQ)**
matryca znaków	**— 9 na 9 punktów (znaki normalne) 9 na 10 punktów (znaki poszerzone)**
wielkość znaków	**— 2,1 na 2,55 mm**
gęstość druku	**— 10 znaków na cal (pica), 12 znaków na cal (elite), 17 znaków na cal (condensed), 5 znaków na cal (pica poszerzony), 6 znaków na cal (elitę poszerzony), 8,5 znaku na cal (condensed poszerzony)**
grafika:	**60, 72, 80, 90 lub 120 punktów na cal**
odstęp wierszy	**— 1/6, 1/8, 7/72, n/72 i n/216 cala**
papier	**— perforowany: 115—254 mm; bez perforacji: 100—245 mm**
interfejs	**— równoległy Centronics (Amphenol) 8-bitowy**
zasilanie	**— 220—240V, 50 Hz**
wymiary	**— długość 400 mm szerokość 250 mm wysokość 100 mm**
masa	**— 4,2 kg**

KOMPUTER W ROLI BAGAŻOWEGO

Czym jest zmora czasami kilkudziesięciominutowego oczekiwania na swój własny bagaż, po wyczerpującym powietrznym rejsie, wiedzą doskonale wszyscy, którzy choć raz znaleźli się w mikroskopijnym w skali największych portów lotniczych świata budyneczku przylotowym warszawskiego Okęcia.

Kufry, nesesery, walizki, torby i pakunki trafiają u nas do samolotu i są z niego wyładowywane nadal dzięki pracy oczu i mięśni silnych mężczyzn w charakterystycznych granatowych drelichach. A przecież bez sprawnego i wydajnego systemu opieki nad tym wszystkim, co podróżni, już we wstępnym kontakcie z obsługą lotniska powierzają jej pieczy, można tylko śnić o wielu milionach pasażerów przybywających do Warszawy i odlatujących z niej z uśmiechem zadowolenia. Nim dowiemy się kto i w jaki sposób będzie projektować nasz nowy port lotniczy — Okęcie II — rzućmy okiem na to, jak przy pomocy najnowszych technologii i wszechwładnej elektroniki z zalewem bagażowym radzą sobie inni.

Bazą Królewskich Holenderskich Linii Lotniczych (KLM) jest lotnisko Schiphol w Amsterdamie. Kiedy spogląda się na fotografie z lat trzydziestych trudno sobie wyobrazić, iż Okęcie i Schiphol wystartowały z tego samego poziomu. Dziś amsterdamskie terminale powietrzne zdaniem wielu ekspertów należą do najsprawniej działających na świecie. O skali ruchu na tym jednym z największych i najruchliwszych portów lotniczych globu świadczy, iż w skali rocznej obsłużyć może od 16 do 18 milionów pasażerów i przeładować 650 tys. ton frachtu. Holendrzy jednak liczą się poważnie z rozszerzeniem sieci połączeń KLM, częstotliwości iotów, a także z naporem innych powietrznych towarzystw korzystających ze Schiphol. Wedle ich szacunków przed rokiem 2000 lotnisko należy przystosować do przyjmowania podwójnej liczby pasażerów i podwojonej masy ładunków.

Zadecydowano więc, że do pracy zaprzęgnie się znacznie bardziej wydajne od ludzi automaty sterowane komputerowo Ich obwodom elektronicznym powierzy się odpowiedzialność za sortowanie i sprawny transport całych gór bagażu. Cały program rozbudowy Schiphol obliczony jest na sumę 1,4 mld florenów. A sam system bagażowy pochłonie 50 milionów florenów. Rusza już w roku bieżącym.

Cała maszyneria umieszczona jest pod ziemią pod płytą lotniska i tzw. "palcem C" portu Schiphol. Układ magazynowy i mechaniczny zajmuje powierzchnię 16 tys. metrów kwadratowych, czyli, jak obliczyli skrupulatni Holendrzy, dwóch stadionów piłkarskich. Pozornie rzecz cała sprowadza się do sieci przenośników, jednak jeśli się weźmie pod uwagę skalę ruchu i specyficzne życzenia pasażerów (niektórzy łapią w Amsterdamie połączenia do różnych zakątków świata, wchodzą i schodzą z pokładów innych samolotów nie należących do KLM) problem nieco się komplikuje.

Wszystko zaczyna się od jednego z 80 stanowisk odprawy bagażowej na Schiphol. Każde zaopatrzone jest w terminal komputerowy i drukarkę nanoszącą dane o podróżnym i jego dobytku, specjalnym kodem na samoprzylepne kawałeczki folii. Bagaż i bilet pasażera opatrzone są w ten sposób przygotowanymi nalepkami. W tym momencie do dzieła przystępują automaty systemu Baggage Handling System. 1300 laserowych czytników od tej pory kontrolować będzie nie tylko układ linii na nalepkach (taki sam jak w przypadku oznaczeń towarów przeznaczonych do sprzedaży w sklepach posiadających elektroniczne kasy), ale także odpowiedni krój czcionek literowego kodu. Wstępna segregacja waliz następuje na jednym z dwóch przenośników głównych o długości 500 m. Później trafią na dalsze 250 mniejszych. W sumie ^300 m specjalnych stale przesuwających się taśm napędzanych silnikami o całkowitej mocy 1600 kW.

Przy każdym zakręcie bagażu system sprawdza zgodność wszystkich zawartych na nalepce informacji z danymi centralnego banku pamięci. Nie ogranicza się więc do przeglądu numeru lotu. Mózgiem potrojonej sieci elektronicznej są cztery centralne komputery Microvax umieszczone w odizolowanym od otoczenia podziemnym pomie- szczeniu. Trzy na zmiany pełniące dyżury i czwarty zapasowy. W przypadku jakiejkolwiek niezgodności "podejrzana" waliza kierowana jest na oddzielny taśmociąg i sprawdzana przez... ludzkie oko i ręce.

Holendrzy szacują, że w ten sposób czas, który pochłania droga bagażu od stanowiska odprawy do luku, czy kontenera bagażowego, skrócony zostanie do 7—8 minut. "Elektroniczni bagażowi" nie męczą się tak łatwo. Pracować mają po 19 godzin dziennie, 365 dni w roku i są tak zaprogramowani, ze praktycznie nie mogą się im zdarzyć pomyłki.

Dla tych ciekawych, którym nie dane będzie osobiście zapoznać się z działaniem komputerów w roli bagażowych przedstawiamy kompletny wydruk na nalepce bagażowej, jaką otrzymałby hipotetyczny Mr Johnson z Atlanty podróżujący liniami KLM z Nowego Jorku przez Amsterdam do Warszawy.

Wojciech Łuczak



WSZYSTKO DLA WSZYSTKICH

Polanglia Ltd

171-175 Uxbridge Road, London W13 9AA
Tel: Londyn 840 1715 Telex: 946581 Fax: 840 7136

NAJNIŻSZE CENY W EUROPIE NA NAJLEPSZY SPRZĘT KOMPUTEROWY

Wyłączne przedstawicielstwo na POLSKĘ firmy

AMSTRAD

Rewelacja Roku — Najlepszy PC na rynku

AMSTRAD PC 1640 ECD

idealny do businessu w pełni zgodny z IBM, maksymalne rozszerzenie skali kolorów [do 64 kolorów], zgodny z EGA, Hercules, MDA i CGA. W składzie mysz, zegar, oba interfejsy i software podobnie jak w PC 1512.

PC 1640 SD MD	Pojedyncza stacja dysków, monochrome monitor	£ 450.-
PC 1640 SD CD	Pojedyncza stacja dysków, kolorowy monitor	£ 580,-
PC 1640 SD ECD	Pojedyncza stacja dysków, kol. monitor wysok Rozdzielczości	£ 700.-
PC 1640 DD MD	Podwójna stacja dysków, monochrome monitor	£ 540.-
PC 1640 DD CD	Podwójna stacja dysków, kolorowy monitor	£ 670.-
PC 1640 DD ECD	Podwójna stacja dysków, koi. monitor wysokiej rozdzielczości	£ 800.-
PC 1640 HD MD	Twardy dysk **20 MB**, monochrome monitor	£ 800.-
PC 1640 HD CD	Twardy dysk **20 MB**, kolorowy monitor	£ 950.-
PC 1640 HD ECD	Twardy dysk **20 MB**, kol. monitor wysokiej rozdzielczości	£ 1080.-

Najpopularniejszy PC Europy:

AMSTRAD PC 1512

(40% rynku brytyjskiego) — **PO ZNIŻONYCH CENACH:**
Zgodny z IBM. w skład wchodzi: mysz, zegar Ouartz, oba interfejsy, software: MSDOS, DOS i GEM z Desktop & Paint, Locomotive BASIC 2.

PC 1512 SD MM	Pojedyncza stacja dysków, monochrome monitor	£ 360.-
PC 1512 DD MM	Podwójna stacja dysków, monochrome monitor	£ 450.-
PC 1512 SD CM	Pojedyncza stacja dysków, kolorowy monitor	£ 500.-
PC 1512 DD CM	Podwójna stacja dysków, kolorowy monitor	£ 600.-
10-DS	uniwersalne dyskietki 5 1/4" D/S D/D [10 sztuk]	£ 10.-
100-DS	uniwersalne dyskietki 5 1/4" D/S D/D [10 pudełek po 10 szt]	£ 55.-

*** *Polanglia Ltd. dodaje bezpłatnie 10 dyskietek 5 1/4 " D/S DAJ przy zakupie* * każdego PC 1512 lub PC 1640 oraz książkę i 6 dysków: Migent/Ability + 4 gry wraz z każdym PC 1512**

RABAT : £ 25.-

przy zakupie PC 1512 lub PC 1640 wraz z rewelacyjną drukarką AMSTRAD DMP 4000 [Drukarka Roku 87], oraz przy zakupie dwóch lub więcej PC na jednego odbiorcę

NAJWYŻSZEJ klasy

DRUKARKI AMSTRAD

po zadziwiająco niskich cenach [z kablem]:
* **Nowa DMP 3160: 160 cps [NLQ 40 cps]** £ 150.-
* **DRUKARKA ROKU: DMP 4000 —** 15", 200 cps [NLQ 50 cps] **100 różnych możliwości druku włącznie z grafiką** £ 240.-
* **od listopada 1987: LQ 3500 — 160 cps [NLQ 50 cps** £ 240.-

AMSTRAD PPC 512/640

w pełni zgodny z IBM; klawiatura odpowiednia do najnowszego standartu PS/2; Stacja dysków 720K 31/3". 640 x 200 pixel supertwist LCD, zgodny z CGA i MDA.

PPC 512 S	Pojedyncza stacja dysków, 720 K	£ 360 .-
PPC 640 S	Pojedyncza stacja dysków, 720 K	£ 450 .-
PPC 512 D	Podwójna stacja dysków, 720 K, wbudowany modem	£ 450 .-
PPC 640 D	Podwójna stacja dysków, 720 K, wbudowany modem	£ 540 .-

Najnowocześniejszy komputer/edytor tekstu z drukarką 'LETTER OUALITY'

AMSTRAD PCW 9512

cena inauguracyjna: £ 475.-
w skład wchodzi: drukarka 15" 'daisywheel' o doskonałej jakości druku [LQ], monitor 90 kolumn, 512 K RAM i napęd dysków 1 MB -i software: LocoScript 2, LocoSpell [słownik angielski], LocoMail.

Na zadanie klientów wznowiono produkcie niezawodnego komputera z edytorem tekstu:

AMSTRAD PCW 8256 i PCW 8512 — po nowej niższej cenie

PCW 8256	komputer 256K, pojedyncza stacja dysków, monitor,drukarka, software	£ 280.-
PCW 8512	komputer 512K, podwójna stacja dysków, monitor, drukarka, software	£ 370.-

Popularna seria komputerów domowych

AMSTRAD CPC 464/6128

po rewelacyjnie niskich cenach:

CPC 464	Z komputer 64K z wbudowanym magnetofonem + zielony monitor	£ 150.-
CPC 464 K	komputer 64K z wbudowanym magnetofonem + kolorowy monitor	£ 220.-
CPC 6128 Z	komputer 128K z wbudowaną stacja dysków + zielony monitor	£ 220.-
CPC 6128 K	komputer 128K z wbudowana stacja dysków + kolorowy monitor	£ 300.-
10-DK	dyskietki 3" [10 sztuk]	£ 25.-
FD-1 + kbl	dodatkowa stacja dysków do CPC 6128 z kablem	£ 100.-
RS 232 C	serial interface do CPC 6128 + software	£ 60.-
MP-2	modulator TV do CPC 6128	£ 30.-
MP-1	modulator TV do CPC 464	£ 15.-
JY-2	joystick do CPC 464 lub CPC 6128	£ 15.-

Nowe SINCLAIR SPECTRUM PLUS 2 i PLUS 3

[produkcja pod kontrolą jakości AMSTRADA]:

SP + 2	komputer 128K z wbudowanym magnetofonem	£ 100.-
JSJ + S/w	joystick i software do SP + 2	£ 20.-
SP + 3 + JSJ	komputer 128K z wbudowaną stacja dysków	£ 190.-

*** W ceny wliczone są:**
wszelkie koszta dewizowe związane z przesyłką, tzn. koszta F.O.B. w Wielkiej Brytanii, opakowanie, ubezpieczenie na transport do Warszawy, Export Licence, itp.
W Polsce zapewniamy serwis na sprzęt AMSTRAD i STAR jedynie zakupiony w firmie Polanglia Ltd.,

Serwis Gwarancyjny:
wykonywany jest w Polsce za pośrednictwem T.E — I. REMEX dostępny za dodatkową opłatę £ 30.- doliczona do zamówienia za każdy komputer AMSTRAD lub system PCW, natomiast £ 15.- za każdą drukarkę, komputery Sinclair i pozostały sprzęt objęty tą ofertą. Serwis pogwarancyjny odpłatny w polskich **złotych, dostępny jest dla wszystkich klientów Polanglii**
Osoby zakupujące sprzęt AMSTRAD w innych firmach eksportowych lub w sklepach nie są uprawnione do korzystania z serwisu AMSTRADA w Polsce.

Jedynie POLANGLIA LTD jest w stanie zapewnić autoryzowany serwis sprzętu komputerowego AMSTRAD w Polsce



Zgodnie z warunkami aktualnej oferty firmy Polanglia, **niniejszym zamawiam:**

.... £ ....
.... £ ....
.... £ ....
PLUS kwota pobierana Przez Barclays Bank = £ 4.-
RAZEM £

Załączam czek lub kopie zlecenia bankowego na przelew w/w sumy na konto Polanglia Ltd Nr. 70736805 w Barclays Bank PLC, Ealing Broadway Branch (kod 20-27-48), 53 The Broadway, LONDON W5 5JS, zrealizowanego w dniu .... / ..... / ...... przez bank .... oddział ....
w ....
Wszelkie tranzakcje podlegają warunkom firmy POLANGLIA opartym na prawie angielskim.
Podpis wpłacającego .... Nazwisko i imię .... Data ....
NAZWISKO I IMIĘ ODBIORCY ....
PEŁNY ADRES ....

	GIEŁDA BAJTKA (tys. zl)	PEWEX BALTONA (USD)	RFN (śred.) (DM)
SINCLAIR			
ZX81	25	–	39
ZX Spectrum 48 kB	90	115	110–150
ZX Spectrum Plus	130	–	180–230
ZX Spectrum 128 kB + 2	220	–	250
Drukarka SEIKOSHA GP50S	85	–	99
TIMEX 2048	130	146	–
Joystick	6	–	4–9
COMMODORE			
C–64	210	219	320
C–128	300	299	480
C–128D	–	–	999
Amiga z monit. kolorowym	1.2 mln	–	3000
Magnetofon 1531	45	48	30
Stacja dyskietek 1541	200	–	399
Stacja dyskietek 1571	240	299	460
Drukarka MPS–803	200	–	149
Dyskietki 5 1/4 (średnia jakość)	0.8–1.5	3.5	0.3–1.5
ATARI			
65 XE	100	199	320
130 XE	140	125	110
Stacja dyskietek 1050	220	187	350
Drukarka 1029	230	199	270–299
Atari 520 STM St. Dysk. 0.5Mb	1.1 mln.	998	800
AMSTRAD			
464 z mon. monochromat.	280	–	400
6128 z mon. monochromat.	430	–	750
6128 z mon. kolor.	530	–	1000
Dyskietki 3"	4–5	–	6-9
Stacja dyskietek 3 do 464	380	–	399
PC 1512	1,2–1,3	–	999

Giełdy Bajtka odbywają się przy ul. Grzybowskiej 35 (Szkota Podstawowa nr 25) w każdą sobotę i niedzielę Od 14.00 do 19.00

Nieco spokojniejsza styczniowa giełda odnotowała spadek cen i dużą podaż sprzętu. Przede wszystkim oferowano wiele mikrokomputerów z rodziny Spectrum będących pod kontrolą jakościową Amstrada. Pojawiły się już pierwsze' egzemplarze Spectrum + 3 z trzy calową stacją dysków. Za tę nowość żądano 400 tys. zł. Natomiast zwolennicy Commodore mieli do dyspozycji kilka mikrokomputerów Amiga 500 i Amiga 1000, ale ich właściciele ograniczali się wyłącznie do wymiany oprogramowania. Obie wersje Amigi wzbudzały ogromne zainteresowanie dzięki rewelacyjnej grafice, przewyższającej rozdzielczością i ilością kolorów nawet najnowszą serię ATARI ST.

(gr)



INDYWIDUALNY BANK DANYCH

Fryderyk Badura, lat 12. Posiada ZX SPECTRUM, kilkanaście gier oraz programów użytkowych. Zainteresowania: filatelistyka i informatyka. Proponuje wymianę programów. Adres: 44-200 Rybnik, ul. Dworcowa 5/41.

Karol Augustyn, lat 18. Posiada mikrokomputer AMSTRAD CPC-464. Proponuje wymianę oprogramowania doświadczeń. Adres: 15-082 Białystok, ul. Nowotki 13/2 m 59.

Piotr Bilski, lat 26, lekarz. Mikrokomputer AMIGA 500, drukarka STAR NL-10, stacja dysków 3,5 "AMIGG. Pragnie nawiązać kontakt z użytkownikami amigi w celu wymiany doświadczeń, oprogramowania oraz literatury. Adres: 70-836 Szczecin, ul. Węgierska 5.

Rafał Gałązka, uczeń 12 lat. Mikrokomputer Atari 800 XL, stacja dysków 1050. Oprogramowanie: kilkadziesiąt gier i programów użytkowych. Proponuje wymianę gier i programów. Adres: 97-300 Piotrków Tryb. ul. Kostromska 66.

Mariusz Fredyn, uczeń lat 14. Posiada mikrokomputer Commodore 64, magnetofon TURBO DATA. Oprogramowanie: programy użytkowe. Proponuje wymianę doświadczeń i programów. Adres: 03-977 Warszawa, ul. Egipska 5 m 14.

Rafał Okarma, uczeń lat 15. Posiada komputer IBM PC wraz z osprzętem. Oprogramowanie: programy prezentacyjne, użytkowe oraz gry. Pragnie nawiązać korespondencję z posiadaczami komputerów tej marki w celu wymiany doświadczeń, literatury oraz oprogramowania. Adres: 33-171 Pleśnią 284, Tarnów.

Przemysław Sadoch, uczeń lat 13. Posiada Atari 65 XE, joystick, magnetofon XC-12. Posiada programy muzyczne oraz matematyczne. Proponuje wymianę oprogramowania. Adres: 64-920 Pita, Pl. Zwycięstwa 10/11.

Dudek Grzegorz, uczeń lat 15. Posiada mikrokomputer Commodore 16/116. Zainteresowania informatyka oraz elektronika. Proponuje wymianę gier oraz programów użytkowych. Adres: 44-100 Gliwice, ul. Lotników 87/29.

Ireneusz Niezgoda, lat 15. Posiada mikrokomputer Commodore Plus 4. Proponuje wymianę oprogramowania. Adres: 05-400 Otwock ul. Batorego 36 m 18.

Robert i Olgierd Zych są posiadaczami mikrokomputera Sharp MZ-800. Pragną nawiązać kontakt z kimś kto zna ten mikrokomputer. Adres: 98-400 Wieruszów, ul. Fabryczna 6 bl 1/43.

Damian Kulak, uczeń lat 19. Posiada Atari 800 XL oraz 130 XE. Pragnie nawiązać kontakt w celu wymiany gier i programów użytkowych. Adres: 72-200 Nowogard, ul. Zamkowa 4/1.

Mariusz Tańczak, lat 14. Posiada mikrokomputer ZX Spectrum 48 KB. Oprogramowanie około 120 gier, programy własne. Proponuje wymianę oprogramowania oraz literatury. Adres: 60-710 Poznań, ul. Kanałowa 15/17.

Drogi Bajtku!

Otrzymałem właśnie w prezencie gwiazdkowym komputer Atari 65XE, przywieziony specjalnie na tę okazję z Zachodu. Jakież było moje zdziwienie, gdy po otwarciu opakowania okazało się, iż jest tam... gra telewizyjna z gniazdem na cartridge'e i dwoma drążkami! Na obudowie widnieje wszakże wyraźnie oznaczenie "Atari 65XE", zresztą obudowa urządzenia stylistycznie przypomina znane mi dobrze z fotografii komputery ST lub XE. Do zestawu dołączone były dwa cartridge'e z grami. Całość działa znakomicie, ja jednak chciałem mieć komputer z prawdziwego zdarzenia, i dlatego pragnę uzyskać odpowiedź, czy posiadane przeze mnie urządzenie jest faktycznie tym samym (w sensie układów elektronicznych), co model 65XE sprzedawany w Pewexie? Gra jest pozbawiona klawiatury; czy możliwe byłoby zatem dołączenie doń klawiatury od "zwykłego" 65XE lub 800XL? Czy wówczas otrzymałbym komputer pełnosprawny, taki jak inne modele firmy Atari? Czy też jest to kolejna "pułapka na oszczędnych", o której "Bajtek" nie zdążył uprzedzić?

A.F.

(nazwisko i adres do wiad. redakcji)

Gra telewizyjna Atari 65XE Game (tak brzmi pełna nazwa) jest stosunkowo nowym produktem Atari, mającym zastąpić na rynku dość stare już podobne urządzenie Atari Video Computer 2600. Idea jest ta sama: komputer bez klawiatury, za to z dwiema manetkami i oprogramowaniem w postaci gier na wymiennych pakietach ROM.

Redakcja "Bajtka" nie miała dotąd styczności z Atari 65XE w takiej postaci. Dysponuję więc tylko materiałami pochodzącymi z pism zagranicznych. Wynika z nich jednak, że do urządzenia tego można dokupić dołączany moduł klawiatury oraz standardowy magnetofon Atari — i wówczas otrzymamy dokładny odpowiednik komputera domowego 65XE.

Nasuwa się przy okazji pewna refleksja. Otóż jeżeli komputer klasy Atari 800XL jest obecnie sprzedawany w formie li tylko gry telewizyjnej — to oznacza to zbliżający się już niechybnie schyłek 8-bitowych komputerów w zastosowaniach amatorskich

Posiadam komputer Timex 2048. Chciałbym dowiedzieć się, jakie różnice zachodzą pomiędzy moim komputerem a ZX Spectrum. Powodem tego pytania jest to, że część programów (np. Uridium) nie chce pracować na moim komputerze. Chcę poznać przyczyny, dlaczego tak się dzieje.

Jakub Husarzewski
ul. Marii Dąbrowskiej 1/62
01-903 Warszawaw

Nie działanie Uridium (i szeregu innych gier) na komputerze Timex spowodowane jest — jak już w "Bajtku" pisaliśmy — zastosowaniem innego układu ULA. Podstawowym problemem jest korzystanie z klawiatury w programach odczytujących jej stan bezpośrednio przy pomocy ULA — bowiem generowane przez ULA Timex-a kody klawiszy są inne niż w oryginalnym komputerze.

Bardziej szczegółowe wyjaśnienia znaleźć można w miesięczniku "Komputer" nr 12/87, str. 21...

Chciałbym podzielić się z Wami moimi wrażeniami dotyczącymi artykułu "System dla ubogich" z listopada 1987. Artykuł ten pokazuje, jak mało jest warta praca ludzi mądrych. Jak trudno jest sprzedać to, co ludzie ci produkują. Uważam, że nie wszystko stracone. Może tak przy okazji warto by było powiedzieć użytkownikom C-64 gdzie i za ile kupię dyskietkę, taśmę lub cartridge i dokumentację do WARSAW BASIC. Czy nie można wydrukować małego kuponu, właśnie w Bajtku o treści

"Kupię WARSAW BASIC.
Nazwisko........................
Imię................................
Adres..............................
Wersja..........................."

tak, jak to robi na str. 26 firma POLANGLIA Ltd. ? Może te smutne twarze doc. dr hab. B. Radziszewskiego i dr inż. K. Gajewskiego byłyby bardziej uśmiechnięte, gdyby od kilku setek użytkowników C-64 wpłynęły zamówienia?

Interesująca mogłaby być książka (skrypt) napisana przez wymienionych panów, przecież to taki ciekawy system. Może u Was ktos będzie chciał pośredniczyć w interesach. Jesteście znani i możecie zarobić parę groszy.

Jurek

Tak, ma Pan całkowitą rację, produkt taki jak WARSAW BASIC powinien już dawno znaleźć się w masowej sprzedaży. Inaczej to jednak wygląda na piśmie, inaczej zaś w realiach doby "powszechnej niemożności", z którą pomimo drugiego etapu reformy, przyjdzie nam się zapewne jeszcze długo zmagać. W przypadku WARSAW BASIC-a sprawa jest tym trudniejsza, że brak jest prawa chroniącego interesy twórców. Nie ma co bowiem ukrywać, że chodzi o zapewnienie w końcu autorom programu i producentowi godziwego zysku. W każdym razie "Bajtek" aktywnie uczestniczy w sprawie WARSAW BASIC-a. Jakie będą tej sprawy wyniki — zobaczymy.

Przy okazji poruszył Pan pewien kapitalny problem, który polecam uwadze wszystkich: kiedy w Polsce uda się komuś wydać popularny program komputerowy RÓWNOCZEŚNIE z podręcznikiem jego obsługi? Dopóki to się nie uda, będziemy stali w miejscu, bo przecież jaka jest różnica pomiędzy piracką kopią programu zagranicznego — bez instrukcji, a powiedzmy, polskim kompilatorem Basica, nabywanym w sklepie drożej — i też bez instrukcji?.

Od dwóch miesięcy mam Amstrada PCW 8256. Bieda w tym, że jest on przystosowany do zasilania o napięciu 110 V. Mam też dwa oryginalne transformatorki, jeden o mocy 50 W, drugi — 1600 W. Ten pierwszy umożliwia pracę PCW przez około 20 minut. Później jest już tak gorący, że trzeba zaprzestać pracy. Ten drugi natomiast się nie nagrzewa, ale chyba podaje zbyt duże napięcie. Co robić?! Czy szukać innych jeszcze urządzeń, czy zdecydować się na przeróbkę całego PCW?

I jeszcze jedno: kupowałem ten komputer z myślą o jego wykorzystaniu naukowym. Chyba dobrze wybrałem? Ale czy to oznacza, że na moim Amstradzie nie można grać? Skąd wziąć literaturę?

Leszek Połomski
ul. Robotnicza IB/26
39-200 Dębica 1

Istnieją dwa sposoby rozwiązania Pańskiego problemu. Pierwszym jest wymiana modułu zasilającego komputera na podobny moduł przeznaczony do europejskich wersji PCW. Wymaga to oczywiście zakupu takiego modułu. Drugim jest wykonanie transformatora 220/110 V, co pozwoliłoby na niedokonywanie zmian w samym komputerze. W każdym wypadku wymaga to współpracy ze specjalistą z dziedziny elektrotechniki, wraz z dokonaniem dokładnej analizy obciążenia projektowanego transformatora.

Komputer wybrał Pan nieźle. Dla maszyn serii PCW pojawiły się również gry, jakkolwiek nie w oszałamiającej ilości.

Z literaturą może Pan mieć sporo kłopotu. Tłumaczeń materiałów zagranicznych i pewnej ilości własnych, polskich opracowań poszukiwać można w firmach prywatnych, zajmujących się dystrybucją oprogramowania. Sprawę częściowo ułatwia fakt, że Amstrad PCW pracuje w systemie CP/M, ale jest szereg programów przeznaczonych tylko dla PCW — i tu zaczynają się trudności.

Posiadam komputer Atari 800XL. Fabryczny BASIC tego komputera, także TURBO BASIC XL nie posiada deklaracji funkcji DEF FN. Jak wprowadzać funkcje do programów?

Juliusz Wiśniewski
ul. Kanałowa 8/15
Poznań

Oczywiście można się obyć bez funkcji. Są dwa sposoby rozwiązywania tego problemu w ATARI BASIC.

Sposób pierwszy. W zwykłym Basic-u najpierw definiujemy funkcję i nadajemy jej pewną nazwę, a potem posługujemy się nią przez użycie tej nazwy. Tak to wygląda na przykład:

```
10 DEF FN A (X) = (X*(X-1))/2
1000 LET Y = 1 + FN A (20)
```

Nasze rozwiązanie polega na pozbyciu się definicji funkcji, a umieszczaniu jej wzoru wszędzie tam, gdzie zachodzi potrzeba wywołania funkcji. W powyższym przykładzie usuwamy linię 10, natomiast linię 1000 modyfikujemy:

```
1000 LET Y = 1 + (20*(20-1))/2
```

Rozwiązanie takie jest zawsze skuteczne. Jego wadą jest większy nakład pracy przy pisaniu programu i zwiększenie jego objętości. Ponieważ jednak w większości interpreterów Basic-a i tak definicje funkcji muszą mieścić się w jednym wierszu — a zatem są z natury krótkie — nie jest to wadą kardynalną.

Sposób drugi. Zamiast definicji funkcji deklarujemy podprogram realizujący te same operacje. Parametry funkcji przekazywać musimy przy pomocy zmiennych. Także wynik funkcji przekazujemy przy pomocy zmiennej (nazwijmy ją WYNIK). Oto przykład:

```
10 LET WYNIK - (X*(X-1))/2
11 RETURN
1000 LET X = 20
1001 GOSUB 10
1002 LET Y = 1 + WYNIK
```

Zauważmy, że wywołanie funkcji wymaga tu użycia aż trzech instrukcji. Jest jednak wada znacznie poważniejsza. Otóż nie możemy w ten sposób zasymulować bezpośrednio np. wywołania:

```
1000 LET Y = 1 f FN A (FN A (6))
```

Wszystkie opisane trudności znikają, gdy dysponujemy Basic-iem z możliwością definiowania procedur z parametrami. Oto już ostatni przykład.

```
10 DEF PROC A (X, WYNIK)
11 LET WYNIK = (X*(X-1))/2
12 END PROC
1000A(20, WYNiK)
1001 LETY = 1+ WYNIKk
```

Marcin Waligórski

WIEŻE I KOPNIĘTY KWADRAT

Cześć Maluchy!

Zgodnie z obietnicą wracamy dzisiaj do naszego nowego przyjaciela, żółwia LOGO. Potrafimy już kierować nim i uczyć go nowych czynności. Spróbujemy więc na początek rysowania kilku figur geometrycznych.

Przyjmiemy jednak pewne założenie — przed rozpoczęciem rysowania żółw powinien znajdować się w dolnej części figury w połowie jej szerokości, po zakończeniu — w górnej. W ten sposób kolejne rysunki umieszczane będą jeden nad drugim.

Rozpoczniemy od zdefiniowania kwadratu:

```
to kwadrat
 rt 90 fd 20
 repeat 3 [lt 90 fd 40]
 lt 90 fd 20
 pu lt 90 fd 40 pd
end
```

Żółw odwraca się w prawo o kąt prosty, maszeruje przed siebie 20 Kroków a następnie powtarza (repeat) trzykrotnie tę samą czynność: obrót w lewo o kąt prosty i przejście do przodu o 40 kroków. Mamy juz więc trzy całe boki kwadratu i jedną połówkę. Robimy obrót w lewo, dwadzieścia kroków do przodu i kwadrat gotowy. Pozostaje jeszcze sprawa jak umieścić żółwia w górnej części rysunku. Najpierw każemy mu "podnieść pisak" (pu). Od tej chwili za poruszającym się żółwiem nie będzie się już ciągnąć rysowana na ekranie kreska. Teraz obrót w lewo, 40 kroków do przodu i możemy znów "opuścić pisak" (pd), potrafiliśmy nauczyć żółwia rysowania kwadratu, z pewnością nie sprawi mu trudności prostokąt:

```
to prostokat
 rt 90 fd 20
 lt 90 fd 20
 lt 90 fd 40
 lt 90 fd 20
 lt 90 fd 20
 pu
 lt 90 fd 20
 pd
end
```

a także trójkąt, równoległobok, romb i trapez:

```
to trojkat
 rt 90 fd 20
 repeat 2 [lt 120 fd 40]
 lt 120 fd 20
 lt 90 pu fd 35
 pd
end


to rownoleglobok
  rt 90 fd 20
  lt 70 fd 20
  lt 110 fd 40
  lt 70 fd 20
  lt 110 fd 20
  pu lt 90 fd 20
 pd
end

to romb
   rt 60 fd 30
   lt 120 fd 30
   lt 60 fd 30
   lt 120 fd 30
   pu
   lt 120 fd 30
   pd
end


to trapez
   rt 90 fd 30
   lt 135 fd 30
   lt 45 fd 20
   lt 60 fd 25
   lt 120 fd 23
   pu
   lt 90 fd 22
   pd
end
```

Nieco inaczej podejdziemy do rysowania okręgu. Ponieważ żółw LOGO nie potrafi poruszać się po liniach krzywych, narysujemy wielobok foremny (to znaczy taki, który ma wszystkie boki i kąty równe) o takiej liczbie boków, że będzie wyglądał jak okrąg:

```
to okrag
   lt 90
   repeat 60 [fd 2 rt 6]
   pu rt 90
   fd 40
   pd

end
```

Nauczmy teraz komputer rysowania wież, które składać się będą z losowo wybranych figur geometrycznych:

```
to wieza.komputera
  pozycja1
  repeat 7
  [make "los random 7
  if:los = 0 [kwadrat]
  if:los = 1 [prostokat]
  if:los = 2 [romb]
  if:los = 3 [rownoleglobok]
  if:los = 4 [trapez]
  if:los = 5 [okrag]
  if:los = 6 [trojkat]]
end
```

polecenie:

make "los random 7 oznacza, że zmienna "los" przyjmuje wartość losową spośród liczb całkowitych 0—6. Następujące po tym instrukcje warunkowe tak bardzo przypominają BASIC, że nie trzeba już chyba nikomu tłumaczyć jaką spełniają funkcję.

Pozostaje jeszcze określić co to jest pozycja 1:

```
to pozycja1
 pu
 home
 lt 90 fd 200
 lt 90 fd 100
 lt 180
pd
end
```

No i proszę! z luźnych klocków powstała nam zabawa w budowanie wież. Najpierw komputer buduje swoją wieżę, a następnie my staramy się stworzyć drugą, identyczną z pierwszą. Robimy to wpisując nazwy kolejnych figur geometrycznych, z których składała się wieża komputera. Przyda się to z pewnością tym, którzy jeszcze ciągle mylą trapez z równoległobokiem i nie mogą sobie przypomnieć jak na serio nazywa się "kopnięty kwadrat".

Najpierw jednak trzeba przenieść żółwia w inne miejsce:

```
to pozycja2
  pu
  home
  rt 90 fd 200
  rt 90 fd 100
  rt 180
  pd
end
```

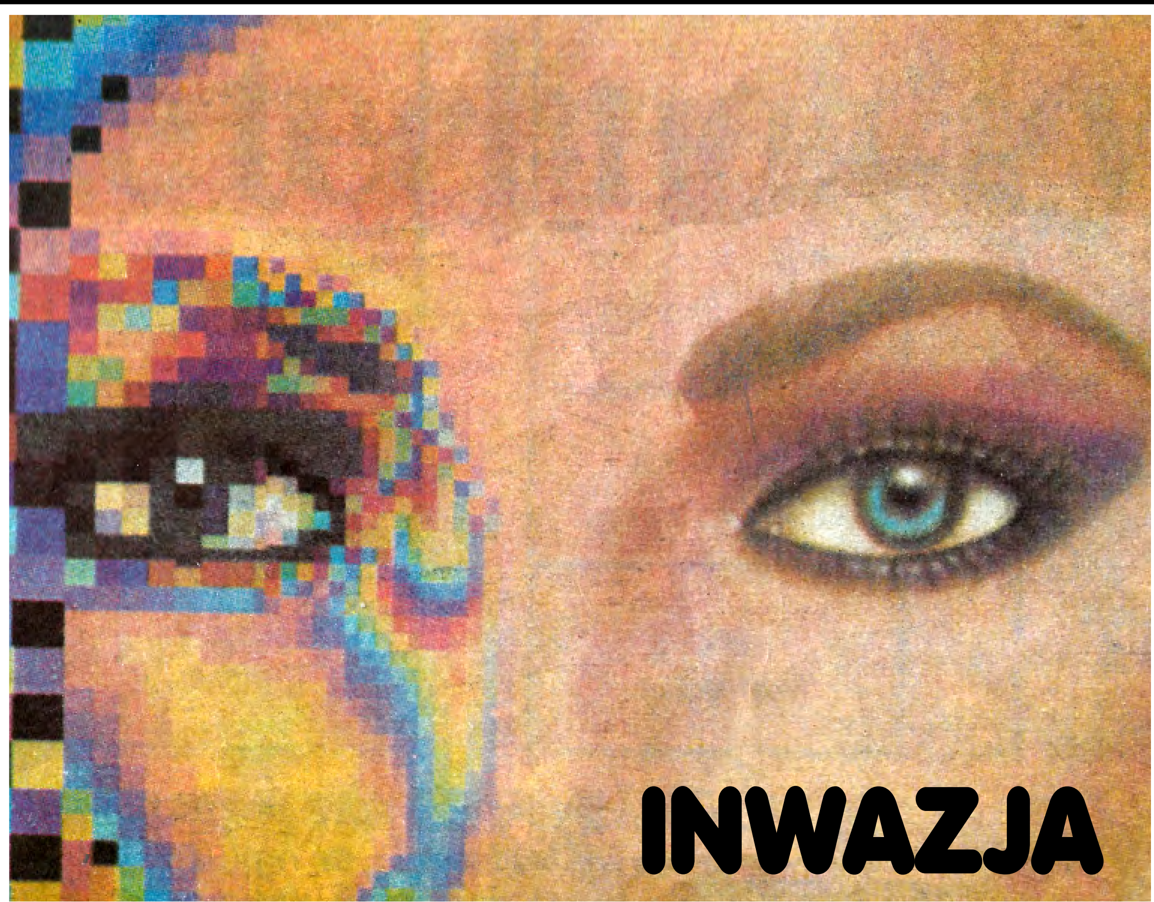
Grę będziemy rozpoczynać od wpisania słowa wieże.

```
to wieze
 CS st
 wieza.komputera
 pozycja2
end
```

Jeśli ktoś z was nie wie jeszcze co to jest "kopnięty kwadrat", to niech weźmie cztery listewki jednakowej długości, zbije je gwoździkami tak, by powstał kwadrat i... kopnie tę konstrukcję. Jeśli listewki nie połamią się i wytrzymają gwoździe, to figura którą zobaczycie będzie właśnie "kopniętym kwadratem".

Nie zniszczcie przy okazji trampek!

Romek



INWAZJA WIDEOFONÓW

Żadna szanująca się rekwizytornia choćby najtańszego i realizowanego najprostszymi środkami filmu science fiction nie może się obyć bez tego jedynego specyficznego sprzętu. Jego obecność w kadrze symbolizuje to. że ludzkość pokonała już dawno pewien etap w historii komunikowania się na odległość i stał się on czymś zupełnie naturalnym w codziennym życiu przyszłych pokoleń.

Marzenia o wideofonie, czyli aparacie stojącym na znacznie wyższym stopniu rozwoju technologicznego niż zwykły telefon, o systemie przekazującym rozmówcy nie tylko dowolne dźwięki, ale również wizerunek tego z kim się porozumiewa, są chyba jeszcze starsze niż wspaniały wynalazek Grahama Bella. Cóż dziś w tym nadzwyczajnego — powiedzą niektórzy — przecież za pośrednictwem sieci telewizyjnej i satelitów możemy uczestniczyć bezpośrednio w wydarzeniach dziejących się tysiące kilometrów od nas, na innych kontynentach. Nawet my w kraju zdystansowanym wyraźnie przez postęp techniki śledziliśmy całkiem na żywo pierwsze godziny szczytu waszyngtońskiego Gorbaczow — Reagan. Bardziej zorientowani dorzucą jeszcze kilka przykładów — telemosty, specjalne międzykontynentalne konferencje prasowe, prywatne narady satelitarne grubych ryb świata międzynarodowego biznesu. To wszystko niesłychanie oszczędza czas dziennikarzy, polityków, czy ludzi interesu, którzy dążąc na odległe miejsce spotkania, musieliby spędzić w kabinach bardziej lub mniej luksusowo wyposażonych samolotów. Nie oszczędza wszakże pieniędzy. Kilkugodzinne połączenie telewizyjne osób komunikujących się za pośrednictwem pojazdu zawieszonego nad Ziemią pochłania bajońskie sumy. Nie możemy więc mówić o powszechności sprawy.

Ową powszechność uzyskamy, kiedy aparat o wymiarach obecnego telefonu podłączony do normalnego przewodu abonenckiego przekaże nam po wybraniu numeru i głos i obraz osoby, z którą pragnęlibyśmy zamienić parę słów, a może pokazać także interesujące ją fotografie, czy też rysunki. Ten wyścig do powszechności wideofonu już się rozpoczął w Stanach Zjednoczonych i Japonii. I to pomimo faktu, że używane w tych krajach linie, zdaniem specjalistów nie bardzo się do tego nadają. A więc trzeba było się dostosować do istniejących sieci.

LU-1000 zamówić może każda firma amerykańska. Firma, bo jest to w tej chwili najdoskonalszy i najdroższy aparat wideofoniczny Ameryki. Kosztuje 1095 dolarów i działa podłączony do najzupełniej normalnego przewodu pod warunkiem, iż pragniemy porozumieć się za jego pomocą z kimś, kto dysponuje drugim LU-1000 zarejestrowanym pod zwykłym numerem telefonicznym. Aparat firmuje usadowiony w Cypress w Kalifornii amerykański oddział słynnego japońskiego konglomeratu Mitsubishi, a sprawą rozwoju wideofonicznej komunikacji zajmuje się świeżo powołana komórka trustu — Visual Telecom Division. Produkcję powierzono należącej do Mitsubishi wytwórni Luma Telecom, stąd nazwa urządzenia LU-1000.

Wideofon Luma składa się z czegoś w rodzaju komputerowej klawiatury, w którą wbudowano głośniki i mikrofony oraz części wizualnej, czyli minikamery video, a także czarnobiałego ekranu telewizyjnego o przekątnej trzech cali. Połączenie następuje poprzez wystukanie nazwiska osoby, z którą pragniemy rozmawiać. LU-1000 zaprogramowany pamięta dane i numery telefoniczne 100 naszych przyjaciół. Mało tego, na żądanie właściciela wyświetla je na swym monitorze.

Kiedy "podniesiemy słuchawkę" aparat pokazuje jedynie ruchomy wizerunek posługującego się nim człowieka, po uzyskaniu kontaktu z drugim abonentem obraz dzieli się na dwoje. Po jednej stronie widzimy osobę z nami dyskutującą, a raczej zapamiętaną i przesłaną nam przez LU-1000 stopklatką z portretem tego, kto podszedł do aparatu, po drugiej w pełni ruchomą własną podobiznę. Tak samo nieruchomo odbierani jesteśmy przez naszego interlokutora. To oczywista niedoskonałość tego pierwszego, powszechnego wideofonu. Ale jego twórcy całą winę zwalają na "zbyt wąskie gardło" w postaci miedzianych przewodów telefonicznych. Jeśli zastąpiłoby się je światłowodami transmisja byłaba absolutnie pełna i żywa, choć nadal w czerni i bieli. Na razie, bo wedle zapewnień producenta to dopiero wprowadzenie do tematu inwazja wideofonów.

Wrażenie ruchu w obrazie naszego rozmówcy w LU-1000 tworzone jest w ten sposób, iż co kilka sekund w zależności od położenia jego twarzy pojawia się nowa stopklatka. Oczywiście można kamerze podsunąć w ten sposób różne przedmioty, pokazać obrazy, plany, czy szkice. To już bardzo wiele jak na początek.

LU-1000 używany jest obecnie w USA przez kilkanaście wielkich przedsiębiorstw, a także przez niektóre oddziały policji. W tej ostatniej roli aparat oddaje nieocenione usługi jako błyskawiczny identyfikator poszukiwanych, Zatrzymanych i podejrzanych o przestępstwo.

Ale amerykańska filia Mitsubishi pomyślała również o tych, których nie stać na skomplikowany wideofon za ponad tysiąc dolarów. Dla nich wprowadzono na rynek również fabrykowany przez Luma Telcom tak zwany konsumencki model taniego i powszechnego aparatu. Nosi nazwę Visual Telephone Display, w skrócie Visitel. Od początku 1988 roku nabyć go można w każdym sklepie prowadzącym tego rodzaju rzeczy, wraz z instrukcją podłączania do gniazdka, za 399 dolarów.

VisiTel wyposażony jest w normalną słuchawkę z mikrofonem, lecz nie pamięta numerów i nazwisk tych, z którymi chętnie rozmawiamy, ma bowiem zwyczajny klawiszowy wybierak. Oczywiście dochodzi do tego zminiaturyzowana kamera i ekran. Ten ostatni większy niż w przypadku LU-1000 — czteroipółcalowy. Odbieramy na nim wyłącznie "zamrożony" obraz tego co zobaczy w pierwszym momencie połączenia kamerą VisiTela, z którym się kontaktujemy.

Tim Beck — kierownik sieci sprzedaży spółki Mitsubishi America zapewnia wszystkich potencjalnych klientów, że VisiTel nie będzie opróżniał kieszeni swoich posiadaczy. Opłaty za wideofoniczną rozmowę mają być podobno identyczne, jak w przypadku zwykłego "ślepego" połączenia. Nie potrzebna jest żadna przystawka, czy konwertor. Tylko sprawne gniazdko sieci telefonicznej.

Z najnowszych informacji wynika, że tacy potentaci branży elektronicznej jak NEC i Panasonic już kupili licencje Mitsubishi i w ciągu 1988 roku zaprezentują własne aparaty wideofo- niczne przeznaczone na rynek amerykański. W Japonii natomiast Sony i Nippon Telegraph and Telephone Corp. (NTT) — największa firma telekomunikacyjna na świecie, wspólnymi siłami starają się przeskoczyć dokonania konkurentów. Ich wideofony, ponoć znacznie doskonalsze, zarówno te dla świata interesu, jak i dla każdego abonenta, spodziewane są już w najbliższych miesiącach w japońskich sklepach.

To wszystko zapowiada kolejny zalew. Teraz na progu ery powszechnego wideofonu pozostało pokonanie jeszcze jednej bariery. Torowanie drogi ku temu, co niedawno występowało jedynie w filmach fantastyczno- -naukowych hamuje miedziany, błyszczący pręt oblany plastykową izolacją. Giganci branży telekomunikacyjnej proponują najbardziej uprzemysłowionym krajom Zachodu 10-letni program przejścia do światłowodów. Projektowany układ oznaczono jako Integrated Services Digital Network (Cyfrowy System Zintegrowanych Usług). W ten sposób wszystko — tekst, rysunki, fotografie barwne i obrazy przesyłane będą mogły być łączami abonenckimi 30 razy szybciej niż obecnie. Mówi się, iż już niedługo będzie można zostawić na ekranie własnego wideofonu odręcznie malowaną kilkoma kolorami wiadomość przeznaczoną tylko dla tego jedynego, który może zadzwonić podczas naszej nieobecności. Na cóż miło się o tym pisze czekając w co najmniej kilkuletniej kolejce na przydział swojego numeru i założenie takiego zwykłego, najzwyklejszego telefonu.

Wojciech Łuczak



www.t2e.pl

Seria „Reduks" ma na celu przetworzenie – pełne zdigitalizowanie prasy komputerowej z początków okresu informatyzacji naszego kraju.

W odróżnieniu od plików w formacie djvu i cbr gdzie format papierowy jest tylko zeskanowany i połączony w jeden dokument, **reduks polega na ponownym przełamaniu dokumentu, wraz z ponowną obróbką graficzną - często z ich ponowną kreacją**. Tam gdzie to możliwe zachowana jest grafika z pierwowzoru, jednak jeśli materiał jest zbyt złej jakości - zostaje on podmieniony (oczywiście jeśli jest dostępny) na równoważny. Jeżeli uda się odnaleźć błąd w zamieszczonych listingach są one poprawiane tak, aby mogły zostać uruchomione na emulatorach lub mikrokomputerach.

Rozpocząłem ten cykl od „Bajtków", jednak mam nadzieję, że uda mi się wzbogacić kolekcję zdigitalizowanych i ponownie przełamanych periodyków także o inne ważne tytuły z lat ubiegłych, o których powoli zapominamy, a bez których współczesny obraz komputeryzacji wyglądałby zgoła inaczej.

Wydania „Reduksowe" są bezpłatne, nie roszczę sobie żadnych praw do zawartości przetwarzanych magazynów, jednak ilość czasu i pracy jaką wkładam aby mogły one trafić w Twoje ręce jest na tyle spora, że pozwolę sobie napomknąć iż możesz wesprzeć mnie przekazując dotację w dowolnej wysokości.

Jeśli posiadasz materiały fotograficzne lub edytorskie, które są lepszej jakości niż prezentowane w numerach, daj znać.



W ramach wolnego czasu odtwarzam stare magazyny komputerowe i prowadzę serwis emulacyjny try2emu. Jeżeli uważasz, że moja publiczna działalność możesz postawić mi kawę, a kawę lubię;)



https://buycoffee.to/t2e

Osoby, które wspierają serwis datkami* mają dostęp do reduksów w wysokiej rozdzielczości z usuniętymi reklamami.

W zależności od wysokości datku każdy otrzymuje czasowy dostęp do reduksów w pełnej rozdzielczości na miesiąc, 3 miesiące lub 6 miesięcy.
Osoby, które aktywnie wspierają projekt mają dostęp nieograniczony.